Complete trace models of state and control (full version)

Guilhem Jaber, Andrzej S. Murawski

Complete trace models of state and control

Guilhem Jaber[1] and Andrzej S. Murawski[2]

[1] Université de Nantes, LS2N CNRS, Inria, France
[2] University of Oxford, UK

Abstract. We consider a hierarchy of four typed call-by-value languages with either higher-order or ground-type references and with either call/cc or no control operator.
Our first result is a fully abstract trace model for the most expressive setting, featuring both higher-order references and call/cc, constructed in the spirit of operational game semantics. Next we examine the impact of suppressing higher-order references and callcc in contexts and provide an operational explanation for the game-semantic conditions known as visibility and bracketing respectively. This allows us to refine the original model to provide fully abstract trace models of interaction with contexts that need not use higher-order references or call/cc. Along the way, we discuss the relationship between error- and termination-based contextual testing in each case, and relate the two to trace and complete trace equivalence respectively.
Overall, the paper provides a systematic development of operational game semantics for all four cases, which represent the state-based face of the so-called semantic cube.

Keywords: contextual equivalence, operational game semantics, higher-order references, control operators

1 Introduction

Research into contextual equivalence has a long tradition in programming language theory, due to its fundamental nature and applicability to numerous verification tasks, such as the correctness of compiler optimisations. Capturing contextual equivalence mathematically, i.e. the *full abstraction* problem [26], has been an important driving force in denotational semantics, which led, among others, to the development of game semantics [2, 12]. Game semantics models computation through sequences of question- and answer-moves by two players, traditionally called O and P, who play the role of the context and the program respectively. Because of its interactive nature, it has often been referred to as a middle ground between denotational and operational semantics.

Over the last three decades the game-semantic approach has led to numerous fully abstract models for a whole spectrum of programming paradigms. Most papers in this strand follow a rather abstract pattern when presenting the models, emphasing structure and compositionality, often developing a correspondence with a categorical framework along the way to facilitate proofs. The operational intuitions behind the games are somewhat obscured in this presentation, and left to be discovered through a deeper exploration of proofs.

In contrast, *operational game semantics* aims to define models in which the interaction between the term and the environment is described through a carefully instrumented labelled transition system (LTS), built using the syntax and operational semantics of the relevant language. Here, the derived trace semantics can be shown to be fully abstract. In this line of work, the dynamics is described more directly and provides operational intuitions about the meaning of moves, while not immediately giving structural insights about the structure of the traces.

In this paper, we follow the operational approach and present a whole hierarchy of trace models for higher-order languages with varying access to higher-order state and control. As a vehicle for our study, we use HOSC, a call-by-value higher-order language equipped with general references and continuations. We also consider its sublanguages GOSC, HOS and GOS, obtained respectively by restricting storage to ground values, by removing continuations, and by imposing both restrictions. We study contextual testing of a class of HOSC terms using contexts from each of the languages $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$; we write $\mathbf{x}$ to refer to each case. Our working notion of convergence will be error reachability, where an error is represented by a free variable. Accordingly, at the technical level, we will study a family of equivalence relations $\cong^{\mathbf{x}}_{err}$, each corresponding to contextual testing with contexts from $\mathbf{x}$, where contexts have the extra power to abort the computation.

Our main results are trace models $\mathbf{Tr_x}(\Gamma \vdash M)$ for each $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$, which capture $\cong^{\mathbf{x}}_{err}$ through trace equivalence:

$$\Gamma \vdash M_1 \cong^{\mathbf{x}}_{err} M_2 \text{ if and only if } \mathbf{Tr_x}(\Gamma \vdash M_1) = \mathbf{Tr_x}(\Gamma \vdash M_2).$$

It turns out that, for contexts with control (i.e. $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}\}$), $\cong^{\mathbf{x}}_{err}$ coincides with the standard notion of contextual equivalence based on termination, written $\cong^{\mathbf{x}}_{ter}$. However, in the other two cases, the former is strictly more discriminating than the latter. We explain how to account for this difference in the trace-based setting, using *complete* traces.

A common theme that has emerged in game semantics is the comparative study of the power of contexts, as it turned out possible to identify combinatorial conditions, namely *visibility* [3] and *bracketing* [22], that correspond to contextual testing in the absence of general references and control constructs respectively. In brief, visibility states that not all moves can be played, but only those that are enabled by a “visible part” of the interaction, which could be thought of as functions currently in scope. Bracketing in turn imposes a discipline on answers, requiring that the topmost question be answered first. In the paper, we provide an operational reconstruction of both conditions.

Overall, we propose a unifying framework for studying higher-order languages with state and control, which we hope will make the techniques of (operational) game semantics clearer to the wider community. The construction of the fully abstract LTSs is by no means automatic, as there is no general methodology for

$$
\begin{array}{lll}
\sigma,\tau & \triangleq & \mathrm{Unit} \mid \mathrm{Int} \mid \mathrm{Bool} \mid \mathrm{ref}\tau \mid \tau\times\sigma \mid \tau\to\sigma \mid \mathrm{cont}\ \tau \\
U,V & \triangleq & () \mid \mathbf{tt} \mid \mathbf{ff} \mid \widehat{n} \mid x \mid \ell \mid \langle U,V\rangle \mid \lambda x^{\tau}.M \mid \mathbf{rec}\ y(x^{\tau}).M \mid \mathrm{cont}_{\tau}\ K \\
M,N & \triangleq & V \mid \langle M,N\rangle \mid \pi_i M \mid MN \mid \mathrm{ref}_{\tau}\ M \mid\ !M \mid M := N \mid \mathrm{if}\ M_1\ M_2\ M_3 \mid M\oplus N \mid M \boxdot N \\
 & & \mid M = N \mid \mathrm{call/cc}_{\tau}(x.M) \mid \mathrm{throw}_{\tau}\ \mathrm{M\ to}\ N \\
K & \triangleq & \bullet \mid \langle V,K\rangle \mid \langle K,M\rangle \mid \pi_i K \mid VK \mid KM \mid \mathrm{ref}_{\tau}\ K \mid\ !K \mid V := K \mid K := M \mid \mathrm{if}\ K\ M\ N \\
 & & \mid K\oplus M \mid V\oplus K \mid K\boxdot M \mid V\boxdot K \mid K = M \mid V = K \mid \mathrm{throw}_{\tau}\ \mathrm{V\ to}\ K \mid \mathrm{throw}_{\tau}\ \mathrm{K\ to}\ M \\
C & \triangleq & \bullet \mid \langle M,C\rangle \mid \langle C,M\rangle \mid \pi_i C \mid \lambda x^{\tau}.C \mid \mathbf{rec}\ y(x^{\tau}).C \mid MC \mid CM \mid \mathrm{ref}_{\tau}\ C \mid\ !C \\
 & & \mid C := M \mid M := C \mid \mathrm{if}\ C\ M\ N \mid \mathrm{if}\ M\ C\ N \mid \mathrm{if}\ M\ N\ C \mid C\oplus M \mid M\oplus C \\
 & & \mid C\boxdot M \mid M\boxdot C \mid C = M \mid M = C \mid \mathrm{call/cc}_{\tau}(x.C) \mid \mathrm{throw}_{\tau}\ \mathrm{C\ to}\ M \mid \mathrm{throw}_{\tau}\ \mathrm{M\ to}\ C
\end{array}
$$

Notational conventions: $x, y \in \mathbf{Var}$, $\ell \in \mathbf{Loc}$, $n \in \mathbb{Z}$, $i \in \{1,2\}$, $\oplus \in \{+,-,*\}$, $\boxdot \in \{=,<\}$
Syntactic sugar: $\mathrm{let}\, x = M \,\mathrm{in}\, N$ stands for $(\lambda x.N)M$ (if x does not occur in N we also write $M;N$)

Fig. 1. HOSC syntax

extracting trace semantics from game models. Some attempts in that direction have been reported in [25], but the type discipline discussed there is far too weak to be applied to the languages we study. As the most immediate precursor to our work, we see the trace model of contextual interactions between HOS contexts and HOS terms from [23]. In comparison, the models developed in this paper are more general, as they consider the interaction between HOSC terms and contexts drawn from any of the four languages ranged over by $\mathbf{x}$.

In the 1990s, Abramsky proposed a research programme, originally called the *semantic cube* [1], which concerned investigating extensions of the purely functional programming language PCF along various axes. From this angle, the present paper is an operational study of a *semantic diamond* of languages with state, with GOS at the bottom, extending towards HOSC at the top, either via GOSC or HOS.

2 HOSC

The main objects of our study will be the language HOSC along with its fragments GOSC, HOS and GOS. HOSC is a higher-order programming language equipped with general references and continuations.

Syntax HOSC syntax is given in Figure 1. Assuming countably infinite sets **Loc** (locations) and **Var** (variables), HOSC typing judgments take the form $\Sigma;\Gamma \vdash M : \tau$, where Σ and Γ are finite partial functions that assign types to locations and variables respectively. We list all the typing rules in the Appendix. In typing judgements, we often write Σ as shorthand for $\Sigma;\emptyset$ (closed) and Γ as shorthand for $\emptyset;\Gamma$ (location-free). Similarly, $\vdash M : \tau$ means $\emptyset;\emptyset \vdash M : \tau$.

$$
\begin{array}{ll|l}
(K[(\lambda x^\sigma.M)V], h) & \to (K[M\{V/x\}], h) & (K[!\ell], h) \to (K[h(\ell)], h) \\
(K[\pi_i\langle V_1, V_2\rangle], h) & \to (K[V_i], h) & (K[\mathrm{ref}\, V], h) \to (K[\ell], h \cdot [\ell \mapsto V]) \\
(K[\mathrm{if}\ \mathbf{tt}\ M_1\ M_2], h) & \to (K[M_1], h) & (K[\ell := V], h) \to (K[()], h[\ell \mapsto V]) \\
(K[\mathrm{if}\ \mathbf{ff}\ M_1\ M_2], h) & \to (K[M_2], h) & (K[\ell = \ell'], h) \to (K[b], h) \\
(K[\widehat{n} \oplus \widehat{m}], h) & \to (K[\widehat{n \oplus m}], h) & \text{with } b = \mathbf{tt} \text{ if } \ell = \ell', \text{otherwise } b = \mathbf{ff} \\
(K[\hat{n} \boxdot \hat{m}], h) & \to (K[b], h) & (K[(\underbrace{\mathbf{rec}\ y(x^\sigma).M}_{U})V], h) \to (K[M\{V/x, U/y\}], h) \\
\multicolumn{2}{l|}{\text{with } b = \mathbf{tt} \text{ if } n \boxdot m, \text{otherwise } b = \mathbf{ff}} & \\
(K[\mathrm{call/cc}(x^\tau.M)], h) & \to (K[M\{\mathrm{cont}_\tau\, K/x\}], h) & (K[\mathrm{throw}_\tau\ \mathrm{V\ to}\ \mathrm{cont}_\tau\, K'], h) \to (K'[V], h)
\end{array}
$$

Fig. 2. Operational reduction for HOSC

Operational semantics A heap h is a finite type-respecting map from **Loc** to values. We write $h : (\Sigma; \Gamma)$, if $\mathrm{dom}(\Sigma) \subseteq \mathrm{dom}(h)$ and $\Sigma; \Gamma \vdash h(\ell) : \sigma$ for $(\ell, \sigma) \in \Sigma$, The operational semantics of HOSC reduces pairs (M, h), where $\Sigma; \Gamma \vdash M : \tau$ and $h : (\Sigma; \Gamma)$. The rules are given in Figure 2, where $\{\cdot\}$ denotes (capture-avoiding) substitution. We write $(M, h) \Downarrow_{ter}$ if there exist V, h' such that $(M, h) \to^* (V, h')$ and V is a value.

We distinguish the following fragments of HOSC.

Definition 1. – GOSC *types are* HOSC *types except that reference types are restricted to* $\mathrm{ref}\iota$*, where* ι *is given by the grammar* $\iota \triangleq \mathrm{Unit} \mid \mathrm{Int} \mid \mathrm{Bool} \mid \mathrm{ref}\iota$*.* GOSC *terms are* HOSC *terms whose typing derivations (i.e. not only the final typing judgments) rely on* GOSC *types only.* GOSC *is a superset of* FOSC *[8], which also includes references to references (the* $\mathrm{ref}\iota$ *case above).*
- HOS *types are* HOSC *types that do not feature the* cont *constructor.* HOS *terms are* HOSC *terms whose typing derivations rely on* HOS *types only. Consequently,* HOS *terms never have subterms of the form* $\mathrm{call/cc}_\tau(x.M)$*,* throw_τ M to N *or* $\mathrm{cont}_\tau\, K$*.*
- GOS *is the intersection of* HOS *and* GOSC*, both for types and terms, i.e. there are no continuations and storage is restricted to values of type* ι*, defined above.*

Definition 2. *Given a* HOSC *term* $\Gamma \vdash M : \tau$*, we refer to types in* Γ *and* τ *as* **boundary types***. Let* $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$*. We say that a* HOSC *term* $\Gamma \vdash M : \tau$ *has an* $\mathbf{x}$ *boundary if all of its boundary types are from* $\mathbf{x}$*.*

Remark 1. Note that typing derivations of HOSC terms with an $\mathbf{x}$ boundary may contain arbitrary HOSC types as long as the final typing judgment uses types from $\mathbf{x}$ only. Consequently, if $\mathbf{x} \neq$ HOSC, HOSC terms with an $\mathbf{x}$ boundary form a strict superset of $\mathbf{x}$.

Next we introduce several notions of contextual testing for HOSC-terms, using various kinds of contexts. For a start, we introduce the classic notion of contextual approximation based on observing termination. The notions are parameterized by $\mathbf{x}$, indicating which language is used to build the testing contexts. We write $\Gamma \vdash C : \tau \to \tau'$ if $\Gamma, x : \tau \vdash C[x] : \tau'$, and $\Gamma \vdash C \div \tau$ if $\Gamma \vdash C : \tau \to \tau'$ for some τ'.

Definition 3 (Contextual Approximation). *Let* $\mathbf{x} \in \{\text{HOSC}, \text{GOSC}, \text{HOS}, \text{GOS}\}$*. Given* HOSC *terms* $\Gamma \vdash M_1, M_2 : \tau$ *with an* $\mathbf{x}$ *boundary, we define* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{ter} M_2$ *to hold, when for all contexts* $\vdash C \div \tau$ *built from the syntax of* $\mathbf{x}$*, if* $(C[M_1], \epsilon) \Downarrow_{ter}$ *then* $(C[M_2], \epsilon) \Downarrow_{ter}$*.*

We also consider another way of testing, based on observing whether a program can reach a breakpoint (error point) inside a context. Technically, the breakpoints are represented as occurrences of a special free error variable err : Unit $\rightarrow$ Unit. Reaching a breakpoint then corresponds to convergence to a stuck configuration of the form $(K[err()], h)$: we write $(M, h) \Downarrow_{err}$ if there exist K, h' such that $(M, h) \rightarrow^* (K[err()], h')$.

Definition 4 (Contextual Approximation through Error). *Let* $\mathbf{x} \in \{$HOSC, FOSC, HOS, GOS$\}$*. Given* HOSC *terms* $\Gamma \vdash M_1, M_2 : \tau$ *with an* $\mathbf{x}$ *boundary and* $err \notin \mathrm{dom}(\Gamma)$*, we define* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{err} M_2$ *to hold, when for all contexts* $err : \text{Unit} \rightarrow \text{Unit} \vdash C \div \tau$ *built from* $\mathbf{x}$*-syntax, if* $(C[M_1], \epsilon) \Downarrow_{err}$ *then* $(C[M_2], \epsilon) \Downarrow_{err}$*.*

For the languages in question, it will turn out that $\lesssim^{\mathbf{x}}_{err}$ is at least as discriminating as $\lesssim^{\mathbf{x}}_{ter}$ for each $\mathbf{x} \in \{\text{HOSC}, \text{GOSC}, \text{HOS}, \text{GOS}\}$, and that they coincide for $\mathbf{x} \in \{\text{HOSC}, \text{GOSC}\}$. We will write $\cong^{\mathbf{x}}_{err}$ and $\cong^{\mathbf{x}}_{ter}$ for the associated equivalence relations.

For higher-order languages with state and control, it is well known that contextual testing can be restricted to evaluation contexts after instantiating the free variables of terms to closed values (the so-called *closed instances of use*, CIU). Let us write $\Sigma, \Gamma' \vdash \gamma : \Gamma$ for substitutions γ such that, for any $(x, \sigma_x) \in \Gamma$, the term $\gamma(x)$ is a value satisfying $\Sigma; \Gamma' \vdash \gamma(x) : \sigma_x$. Then $M\{\gamma\}$ stands for the outcome of applying γ to M.

Definition 5 (CIU Approximation). *Let* $\mathbf{x} \in \{\text{HOSC}, \text{GOSC}, \text{HOS}, \text{GOS}\}$ *and let* $\Gamma \vdash M_1, M_2 : \tau$ *be* HOSC *terms with an* $\mathbf{x}$ *boundary.*

- $\Gamma \vdash M_1 \lesssim^{\mathbf{x}(ciu)}_{ter} M_2 : \tau$*, when for all* Σ, h, K, γ*, all built from* $\mathbf{x}$ *syntax, such that* $h : \Sigma$*,* $\Sigma \vdash K \div \tau$*, and* $\Sigma \vdash \gamma : \Gamma$*, we have* $(K[M_1\{\gamma\}], h) \Downarrow_{ter}$ *implies* $(K[M_2\{\gamma\}], h) \Downarrow_{ter}$*.*
- *We write* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}(ciu)}_{err} M_2 : \tau$*, when for all* Σ, h, K, γ*, all built from* $\mathbf{x}$ *syntax, such that* $h : \Sigma; \widehat{err}$*,* $\Sigma; \widehat{err} \vdash K \div \tau$*, and* $\Sigma; \widehat{err} \vdash \gamma : \Gamma$*, we have* $(K[M_1\{\gamma\}], h) \Downarrow_{err}$ *implies* $(K[M_2\{\gamma\}], h) \Downarrow_{err}$*, where* $err \notin \mathrm{dom}(\Gamma)$ *and* $\widehat{err}$ *stands for* $err : \text{Unit} \rightarrow \text{Unit}$*.*

Results stating that "CIU tests suffice" are referred to as CIU lemmas. A general framework for obtaining such results for higher-order languages with effects was developed in [10, 33]. The results stated therein are for termination-based testing, i.e. $\Downarrow_{ter}$, but adapting them to $\Downarrow_{err}$ is not problematic.

Lemma 1 (CIU Lemma). *Let* $\mathbf{x} \in \{\text{HOSC}, \text{GOSC}, \text{HOS}, \text{GOS}\}$ *and* $\mathbf{y} \in \{ter, err\}$*. Then we have* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{\mathbf{y}} M_2$ *iff* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}(ciu)}_{\mathbf{y}} M_2$*.*

The preorders $\lesssim^{\mathbf{x}}_{err}$ will be the central object of study in the paper. Among others, we shall provide their alternative characterizations using trace semantics.The characterizations will apply to a class of terms that we call *cr-free*.

Definition 6. *A* HOSC *term* $\Gamma \vdash M : \tau$ *is* **cr-free** *if it does not contain occurrences of* $\mathrm{cont}_\sigma K$ *and locations, and its boundary types are* cont- *and* ref*-free.*

We stress that the boundary restriction applies to Γ and τ only, and subterms of M may well contain arbitrary HOSC types and occurrences of ref_σ, $\mathrm{call/cc}_\sigma$, throw_σ for any σ. The majority of HOSC/GOSC/HOS/GOS examples studied in the literature, e.g. [28, 4, 8], are actually cr-free. We will revisit some of them as Examples 6, 7, 10. The fact that cr-free terms may not contain subterms $\mathrm{cont}_\tau K$ or ℓ is not really a restriction, as $\mathrm{cont}_\tau K$ and ℓ being more of a run-time construct than a feature meant to be used directly by programmers. Finally, we note that the boundary of a cr-free term is an $\mathbf{x}$ boundary for any $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$. Thus, we can consider approximation between cr-terms for any $\mathbf{x}$ from the range, i.e. the notions $\lesssim^{\mathbf{x}}_{err}$, $\lesssim^{\mathbf{x}}_{ter}$ are all applicable. Consequently, cr-free terms provide a common setting in which the discriminating power of HOSC, GOSC, HOS and GOS contexts can be compared. We discuss the scope for extending our results outside of the cr-free fragment, and for richer type systems, in Section 7.

3 HOSC[HOSC]

Recall that $\lesssim^{\mathrm{HOSC}}_{err}$ concerns testing HOSC terms with HOSC contexts. Accordingly, we call this case HOSC[HOSC]. For $\mathrm{cont}_\sigma(K)$-free terms, we show that $\lesssim^{\mathrm{HOSC}}_{err}$ and $\lesssim^{\mathrm{HOSC}}_{ter}$ coincide, which follows from the lemma below.

Lemma 2. *Suppose* $\Gamma \vdash M_1, M_2$ *be* HOSC *terms not containing any occurrences of* $\mathrm{cont}_\tau(K)$.

1. $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{err} M_2$ *implies* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{ter} M_2$, *for* $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$.
2. $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{ter} M_2$ *implies* $\Gamma \vdash M_1 \lesssim^{\mathbf{x}}_{err} M_2$, *for* $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}\}$.

In what follows, after introducing several preliminary notions, we shall design a labelled transition system (LTS) whose traces will turn out to capture contextual interactions involved in testing cr-free terms according to $\lesssim^{\mathrm{HOSC}}_{err}$. This will enable us to capture $\lesssim^{\mathrm{HOSC}}_{err}$ via trace inclusion. Actions of the LTS will refer to functions and continuations in a symbolic way, using typed names.

3.1 Names and abstract values

Definition 7. *Let* $\mathrm{FNames} = \biguplus_{\sigma,\sigma'} \mathrm{FNames}_{\sigma\to\sigma'}$ *be the set of* **function names**, *partitioned into mutually disjoint countably infinite sets* $\mathrm{FNames}_{\sigma\to\sigma'}$. *We will use* f, g *to range over* FNames , *and write* $f : \sigma \to \sigma'$ *for* $f \in \mathrm{FNames}_{\sigma\to\sigma'}$.

Analogously, let $\mathrm{CNames} = \biguplus_\sigma \mathrm{CNames}_\sigma$ *be the set of* **continuation names**. *We will use* c, d *to range over* CNames, *and write* $c : \sigma$ *for* $c \in \mathrm{CNames}_\sigma$. *Note* *that the constants represent continuations, so the "real" type of c is* cont σ*, but we write $c : \sigma$ for the sake of brevity. We assume that* CNames, FNames *are disjoint and let* Names = FNames $\uplus$ CNames*. Elements of* Names *will be weaved into various constructions in the paper, e.g. terms, heaps, etc. We will then write $\nu(X)$ to refer to the set of names used in some entity X.*

Because of the shape of boundary types in cr-free terms and, in particular, the presence of product types, the values that will be exchanged between the context and the program take the form of tuples consisting of (), integers, booleans and functions. To describe such scenarios, we introduce the notion of ***abstract values***, which are patterns that match such values. Abstract values are generated by the grammar

$$A, B \triangleq () \mid \mathbf{tt} \mid \mathbf{ff} \mid \widehat{n} \mid f \mid \langle A, B \rangle$$

with the proviso that, in any abstract value, a name may occur at most once. As function names are intrinsically typed, we can assign types to abstract values in the obvious way, writing $A : \tau$.

3.2 Actions and traces

Our LTS will be based on four kinds of actions, listed below. Each action will be equipped with a ***polarity***, which is either Player (P) or Opponent (O). P-actions describing interaction steps made by a tested term, while O-actions involve the context.

- ***Player Answer*** (PA) $\bar{c}(A)$, where $c : \sigma$ and $A : \sigma$. This action corresponds to the term sending an abstract value A through a continuation name c.
- ***Player Question*** (PQ) $\bar{f}(A, c)$, where $f : \sigma \to \sigma'$, $A : \sigma$ and $c : \sigma'$. Here, an abstract value A and a continuation name c are sent by the term through a function name f.
- ***Opponent Answer*** (OA) $c(A)$, $c : \sigma$ then $A : \sigma$. In this case, an abstract value A is received from the environment via the continuation name c.
- ***Opponent Question*** (OQ) $f(A, c)$, where $f : \sigma \to \sigma'$, $A : \sigma$ and $c : \sigma'$. Finally, this action corresponds to receiving an abstract value A and a continuation name c from the environment through a function name f.

In what follows, $\mathbf{a}$ is used to range over actions. We will say that a name is ***introduced*** by an action $\mathbf{a}$ if it is sent or received in $\mathbf{a}$. If $\mathbf{a}$ is an O-action (resp. P-action), we say that the name was introduced by O (resp. P). An action $\mathbf{a}$ is ***justified*** by another action $\mathbf{a}'$ if the name that $\mathbf{a}$ uses to communicate, i.e. f in questions ($\bar{f}(A, c)$, $f(A, c)$) and c in answers ($\bar{c}(A)$, $c(A)$), has been introduced by $\mathbf{a}'$.

We will work with sequences of actions of a very special shape, specified below. The definition assumes two given sets of names, N_P and N_O, which represent names that have already been introduced by P and O respectively.

Definition 8. *Let $N_O, N_P \subseteq$* Names*. An (N_O, N_P)-**trace** is a sequence t of actions such that:*

- *the actions alternate between Player and Opponent actions;*
- *no name is introduced twice;*
- *names from N_O, N_P need no introduction;*
- *if an action* $\mathbf{a}$ *uses a name to communicate then*
  - $\mathbf{a} = \bar{f}(A,c)$ *($f \in N_O$) or* $\mathbf{a} = \bar{c}(A)$ *($c \in N_O$) or* $\mathbf{a} = f(A,c)$ *($f \in N_P$) or* $\mathbf{a} = c(A)$ *($c \in N_P$) or*
  - *the name has been introduced by an earlier action* $\mathbf{a}'$ *of opposite polarity.*

Note that, due to the shape of actions, a continuation name can only be introduced/justified by a question. Moreover, because names are never introduced twice, if $\mathbf{a}'$ justifies $\mathbf{a}$ then $\mathbf{a}'$ is uniquely determined in a given trace. Readers familiar with game semantics will recognize that traces are very similar to alternating justified sequences except that traces need not be started by O.

Example 1. Let $(N_O, N_P) = (\{c\}, \emptyset)$ where $c : \tau = ((\mathrm{Unit} \to \mathrm{Unit}) \to \mathrm{Unit}) \times (\mathrm{Unit} \to \mathrm{Int})$. Then the following sequence is an (N_O, N_P)-trace:

$$\mathtt{t}_1 = \bar{c}(\langle g_1, g_2\rangle)\ \ g_1(f_1, c_1)\ \ \bar{f_1}((), c_2)\ \ c_2(())\ \ \bar{c_1}(())\ \ c_2(())\ \ \bar{c_1}(())\ \ g_2((), c_3)\ \ \bar{c_3}(2)$$

where $g_1 : (\mathrm{Unit} \to \mathrm{Unit}) \to \mathrm{Unit}$, $g_2 : \mathrm{Unit} \to \mathrm{Int}$, $f_1 : \mathrm{Unit} \to \mathrm{Unit}$, $c_1, c_2 :$ Unit, $c_3 : \mathrm{Int}$.

3.3 Extended syntax and reduction

We extend the definition of HOSC presented in Figure 2 to take into account these names. We refine the operational reduction using continuation names to keep track of the toplevel continuation. We list all the changes below.

- Function names are added to the syntax as *constants*. Since they are meant to represent values, they are also considered to be syntactic values in the extended language.
$$\frac{f \in \mathrm{FNames}_{\sigma \to \sigma'}}{\Sigma; \Gamma \vdash f : \sigma \to \sigma'}$$
- Continuation names are *not* terms on their own. Instead, they are built into the syntax via a new construct $\mathrm{cont}_\sigma\,(K, c)$, subject to the following typing rule.
$$\frac{\Sigma; \Gamma \vdash K : \sigma \to \sigma' \quad c \in \mathrm{CNames}_{\sigma'}}{\Sigma; \Gamma \vdash \mathrm{cont}_\sigma\,(K, c) : \mathrm{cont}\ \sigma}$$
  $\mathrm{cont}_\sigma\,(K, c)$ is a staged continuation that first evaluates terms inside K and, if this produces a value, the value is passed to c. This operational meaning will be implemented through a suitable reduction rule, to be discussed next. $\mathrm{cont}_\sigma\,(K, c)$ is also regarded as a value. Note that we remove the old construct $\mathrm{cont}_\sigma\, K$ from the extended syntax.
- The operational semantics $\to$ underpinning the LTS is based on triples (M, c, h) such that $\Sigma; \Gamma \vdash M : \sigma$, $c \in \mathrm{CNames}_\sigma$ and $h : \Sigma$. The continuation name c is used to represent the surrounding context, which is left abstract.

The previous operational rules $\rightarrow$ are embedded into the new reduction $\rightarrow$ using the rule below.

$$\frac{(M,h)\rightarrow(M',h')}{(M,c,h)\rightarrow(M',c,h')}$$

The two reduction rules related to continuations, previously used to define $\rightarrow$, are *not* included. Instead we use the following rules, which take advantage of the extended syntax.

$$\begin{array}{rl}(K[\mathrm{call/cc}_\tau(x.M)],c,h) &\rightarrow (K[M\{\mathrm{cont}_\tau\,(K,c)/x\}],c,h)\\ (K[\mathrm{throw}_\tau\ \mathrm{V}\ \mathrm{to}\ \mathrm{cont}_\tau\,(K',c')],c,h) &\rightarrow (K'[V],c',h)\end{array}$$

3.4 Configurations

We write Vals for the extended set of syntactic values, i.e. FNames $\subseteq$ Vals. Let ECtxs stand for the set of extended evaluation contexts, defined as K in Figure 1 taking the extended definition of values into account. Before defining the transition relation of our LTS, we discuss the shape of configurations, providing intuitions behind each component.

Passive configurations take the form $\langle\gamma,\xi,\phi,h\rangle$ and are meant to represent stages at which the environment is to make a move.

- $\gamma:(\mathrm{FNames}\rightharpoonup\mathrm{Vals})\uplus(\mathrm{CNames}\rightharpoonup\mathrm{ECtxs})$ is a finite map. It will play the role of an environment that relates function names communicated to the environment (i.e. those introduced by P) to syntactic values, and continuation names introduced by P to evaluation contexts.
- $\xi:(\mathrm{CNames}\rightharpoonup\mathrm{CNames})$ is a finite map. It complements the role of γ for continuation names and indicates the continuation to which the outcome of applying $\gamma(c)$ should be passed.
- $\phi\subseteq$ Names. The set ϕ will be used to collect all the names used in the interaction, regardless of which participant introduced them. Following our description above, those introduced by O will correspond to $\phi\setminus\mathrm{dom}(\gamma)$.

The components satisfy healthiness conditions, implied by their role in the system. Let $\Sigma=\mathrm{dom}(h)$.

- If $f:\mathrm{dom}(\gamma)\cap\mathrm{FNames}_{\sigma\rightarrow\sigma'}$ then $\gamma(f)$ is a value such that $\Sigma\vdash\gamma(f):\sigma\rightarrow\sigma'$.
- $\mathrm{dom}(\xi)=\mathrm{dom}(\gamma)\cap\mathrm{CNames}$.
- If $c:\mathrm{dom}(\gamma)\cap\mathrm{CNames}_\sigma$ and $\Sigma\vdash\gamma(c):\sigma\rightarrow\sigma'$ then $\xi(c)\in\mathrm{CNames}_{\sigma'}$.
- Finally, names introduced by the environment and communicated to the program may end up in the environments and the heap: $\nu(\mathrm{img}(\gamma)),\nu(\mathrm{img}(\xi))$, $\nu(\mathrm{img}(h))\subseteq\phi\setminus\mathrm{dom}(\gamma)$.

Active configurations take the form $\langle M,c,\gamma,\xi,\phi,h\rangle$ and represent interaction steps of the term. The γ,ξ,ϕ,h components have already been described above. For M and c, given $\Sigma=\mathrm{dom}(h)$, we will have $\Sigma;\emptyset\vdash M:\sigma$, $c\in\mathrm{CNames}_\sigma$ and $\nu(M)\cup\{c\}\subseteq\phi\setminus\mathrm{dom}(\gamma)$.

3.5 Transitions

Observe that any closed value V of a cont- and ref-free type σ can be decomposed into an abstract value A (pattern) and the corresponding substitution γ (matching). The set of all such decompositions, written $\mathbf{AVal}_\sigma(V)$, is defined below. Given a value V of a (cr-free) type σ, $\mathbf{AVal}_\sigma(V)$ contains all pairs (A, γ) such that A is an abstract value and $\gamma : \nu(A) \to \mathrm{Vals}$ is a substitution such that $A\{\gamma\} = V$. More concretely,

$$
\begin{array}{ll}
\mathbf{AVal}_\sigma(V) & \triangleq \{(V, \emptyset)\} \quad \text{for } \sigma \in \{\mathrm{Unit}, \mathrm{Bool}, \mathrm{Int}\} \\
\mathbf{AVal}_{\sigma\to\sigma'}(V) & \triangleq \{(f, [f \mapsto V]) \mid f \in \mathrm{FNames}_{\sigma\to\sigma'}\} \\
\mathbf{AVal}_{\sigma\times\sigma'}(\langle U, V\rangle) & \triangleq \{(\langle A_1, A_2\rangle, \gamma_1 \cdot \gamma_2) \mid (A, \gamma_1) \in \mathbf{AVal}_\sigma(U),\ (A_2, \gamma_2) \in \mathbf{AVal}_{\sigma'}(V)\}
\end{array}
$$

Note that, by writing $\cdot$, we mean to implicitly require that the function domains be disjoint. Similarly, when writing $\uplus$, we stipulate that the argument sets be disjoint.

Example 2. Let $\sigma = (\mathrm{Int} \to \mathrm{Bool}) \times (\mathrm{Int} \times (\mathrm{Unit} \to \mathrm{Int}))$ and $V \equiv \langle \lambda x^{\mathrm{Int}}.x \neq 1, \langle 2, \lambda x^{\mathrm{Unit}}.3\rangle\rangle$. Then $\mathbf{AVal}_\sigma(V)$ equals

$$\{(\langle f, \langle 2, g\rangle\rangle, [f \mapsto (\lambda x^{\mathrm{Int}}.x \neq 1)]\cdot[g \mapsto (\lambda x^{\mathrm{Unit}}.3)]) \ \mid\ f \in \mathrm{FNames}_{\mathrm{Int}\to\mathrm{Unit}},\ g \in \mathrm{FNames}_{\mathrm{Unit}\to\mathrm{Int}}\}.$$

Finally, we present the transitions of, what we call the HOSC[HOSC] LTS, in Figure 3.

Example 3. We analyze the (PQ) rule below in more detail.

$$
\begin{array}{l}
(PQ)\ \langle K[fV], c, \gamma, \xi, \phi, h\rangle \xrightarrow{\bar{f}(A,c')} \langle \gamma \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \phi \uplus \nu(A) \uplus \{c'\}, h\rangle \\
\qquad\quad \text{when } f : \sigma \to \sigma',\ (A, \gamma') \in \mathbf{AVal}_\sigma(V) \text{ and } c' : \sigma'
\end{array}
$$

The use of $\uplus$ in $\phi \uplus \nu(A) \uplus \{c'\}$ is meant to highlight the requirement that the names introduced in $\bar{f}(A, c')$, i.e. $\nu(A) \cup \{c'\}$, should be fresh and disjoint from ϕ. Moreover, note how γ and ξ are updated. In general, γ, ξ, h are updated during P-actions.

Definition 9. *Given two configurations $\mathbf{C}, \mathbf{C}'$, we write $\mathbf{C} \stackrel{\mathbf{a}}{\Longrightarrow} \mathbf{C}'$ if $\mathbf{C} \xrightarrow{\tau}{}^{*} \mathbf{C}'' \xrightarrow{\mathbf{a}} \mathbf{C}'$, with $\xrightarrow{\tau}{}^{*}$ representing multiple (possibly none) τ-actions. This notation is extended to sequences of actions: given $\mathtt{t} = \mathbf{a}_1 \ldots \mathbf{a}_n$, we write $\mathbf{C} \stackrel{\mathtt{t}}{\Longrightarrow} \mathbf{C}'$, if there exist $\mathbf{C}_1, \ldots, \mathbf{C}_{n-1}$ such that $\mathbf{C} \stackrel{\mathbf{a}_1}{\Longrightarrow} \mathbf{C}_1 \cdots \mathbf{C}_{n-1} \stackrel{\mathbf{a}_n}{\Longrightarrow} \mathbf{C}'$. We define $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C}) = \{\mathtt{t} \mid \text{there exists } \mathbf{C}' \text{ such that } \mathbf{C} \stackrel{\mathtt{t}}{\Longrightarrow} \mathbf{C}'\}$.*

Lemma 3. *Suppose $\mathbf{C} = \langle \gamma, \xi, \phi, h\rangle$ or $\mathbf{C} = \langle M, c, \gamma, \xi, \phi, h\rangle$ are configurations. Then elements of $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ are $(\phi \setminus \mathrm{dom}(\gamma), \mathrm{dom}(\gamma))$-traces.*

$$
\begin{array}{l|lll}
(P\tau) & \langle M, c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\tau} & \langle N, c', \gamma, \xi, \phi, h'\rangle \\
 & \multicolumn{3}{l}{\text{when } (M, c, h) \to (N, c', h')} \\
(PA) & \langle V, c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\bar{c}(A)} & \langle \gamma \cdot \gamma', \xi, \phi \uplus \nu(A), h\rangle \\
 & \multicolumn{3}{l}{\text{when } c : \sigma,\ (A, \gamma') \in \mathbf{AVal}_\sigma(V)} \\
(PQ) & \langle K[fV], c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\bar{f}(A,c')} & \langle \gamma \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \phi \uplus \nu(A) \uplus \{c'\}, h\rangle \\
 & \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma',\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ c' : \sigma'} \\
(OA) & \langle \gamma, \xi, \phi, h\rangle & \xrightarrow{c(A)} & \langle K[A], c', \gamma, \xi, \phi \uplus \nu(A), h\rangle \\
 & \multicolumn{3}{l}{\text{when } c : \sigma,\ A : \sigma,\ \gamma(c) = K,\ \xi(c) = c'} \\
(OQ) & \langle \gamma, \xi, \phi, h\rangle & \xrightarrow{f(A,c)} & \langle VA, c, \gamma, \xi, \phi \uplus \nu(A) \uplus \{c\}, h\rangle \\
 & \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma',\ A : \sigma,\ c : \sigma',\ \gamma(f) = V}
\end{array}
$$

NB $c : \sigma$ stands for $c \in \mathrm{CNames}_\sigma$.

Fig. 3. HOSC[HOSC] LTS

M_1^{cwl}	M_2^{cwl}
M_1^{cwl} : let x = ref 0 in let b = ref **ff** in ⟨λf. if ¬(!b) then b := **tt**; f(); x :=!x + 1; b := **ff**; else (), λ_ : Unit.!x⟩	M_2^{cwl} : let x = ref 0 in let b = ref **ff** in ⟨λf. if ¬(!b) then b := **tt**; let n =!x in f(); x := n + 1; b := **ff**; else () , λ_ : Unit.!x⟩

Fig. 4. Callback-with-lock Example [4]

Example 4. In Figure 5, we show that the trace from Example 1 is generated by the configuration $\mathbf{C} \triangleq \langle M_1^{cwl}, c, \emptyset, \emptyset, \{c\}, \emptyset\rangle$, where M_1^{cwl} is given in Figure 4. We write inc $\triangleq \lambda f.\text{if } \neg(!\ell_b)\ (\ell_b := \mathbf{tt}; f(); \ell_x :=!\ell_x + 1; \ell_b := \mathbf{ff})\ ()$, get $\triangleq \lambda\_.!\ell_x$ and $c : ((\mathrm{Unit} \to \mathrm{Unit}) \to \mathrm{Unit}) \times (\mathrm{Unit} \to \mathrm{Int})$. It is interesting to notice that in this interaction, Opponent uses the continuation N twice, incrementing the counter x by two. The second time, it does it without having to call inc again, but rather by using the continuation name c_2.

Remark 2. Due to the freedom of name choice, note that $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ is closed under type-preserving renamings that preserve names from $\mathbf{C}$.

3.6 Correctness and full abstraction

We define two kinds of special configurations that will play an important role in spelling out correctness results for the HOSC[HOSC] LTS. Let $\Gamma = \{x_1 : \sigma_1, \cdots, x_k : \sigma_k\}$. A map ρ from $\{x_1, \cdots, x_k\}$ to the set of abstract values will be called a ***Γ-assignment*** provided, for all $1 \leq i \neq j \leq k$, we have $\rho(x_i) : \sigma_i$ and $\nu(\rho(x_i)) \cap \nu(\rho(x_j)) = \emptyset$.

$$
\begin{array}{rll}
\mathbf{C} = & \langle M_1^{cwl}, c, \emptyset, \emptyset, \{c\}, \emptyset\rangle & \\
\xrightarrow{\tau^*} & \langle\langle \text{inc}, \text{get}\rangle, c, \emptyset, \emptyset, \{c\}, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 0]\rangle & \\
\xrightarrow{\bar{c}(\langle g_1, g_2\rangle)} & \langle \gamma_1, \emptyset, \{c, g_1, g_2\}, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 0]\rangle & \text{with } \gamma_1 = [g_1 \mapsto \text{inc}, g_2 \mapsto \text{get}], \\
\xrightarrow{g_1(f_1, c_1)} & \langle \text{inc} f_1, c_1, \gamma_1, \emptyset, \phi_2, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 0]\rangle & \text{with } \phi_2 = \{c, g_1, g_2, f_1, c_1\} \\
\xrightarrow{\tau^*} & \langle f_1(); N, c_1, \gamma_1, \emptyset, \phi_2, [\ell_b \mapsto \mathbf{tt}, \ell_x \mapsto 0]\rangle & \text{with } N = \ell_x := !\ell_x + 1; \ell_b := \mathbf{ff} \\
\xrightarrow{\bar{f}_1((), c_2)} & \langle \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{tt}, \ell_x \mapsto 0]\rangle & \text{with } \gamma_2 = \gamma_1 \cdot [c_2 \mapsto \bullet; N], \\
\xrightarrow{c_2(())} & \langle (); N, c_1, \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{tt}, \ell_x \mapsto 0]\rangle & \quad \xi = [c_2 \mapsto c_1] \text{ and } \phi_3 = \phi_2 \uplus \{c_2\} \\
\xrightarrow{\tau^*} & \langle (), c_1, \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 1]\rangle & \\
\xrightarrow{\bar{c_1}(())} & \langle \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 1]\rangle & \\
\xrightarrow{c_2(())} & \langle (); N, c_1, \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 1]\rangle & \\
\xrightarrow{\tau^*} & \langle (), c_1, \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 2]\rangle & \\
\xrightarrow{\bar{c_1}(())} & \langle \gamma_2, \xi, \phi_3, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 2]\rangle & \\
\xrightarrow{g_2((), c_3)} & \langle \text{get}(), c_3, \gamma_2, \xi, \phi_4, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 2]\rangle & \text{with } \phi_4 = \phi_3 \uplus \{c_3\} \\
\xrightarrow{\tau^*} & \langle 2, c_3, \gamma_2, \xi, \phi_4, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 2]\rangle & \\
\xrightarrow{\bar{c_3}(2)} & \langle \gamma_2, \xi, \phi_4, [\ell_b \mapsto \mathbf{ff}, \ell_x \mapsto 2]\rangle &
\end{array}
$$

Fig. 5. Trace derivation in the HOSC[HOSC] LTS

Definition 10 (Program configuration). *Given a Γ-assignment ρ, a cr-free HOSC term $\Gamma \vdash M : \tau$ and $c : \tau$, we define the active configuration $\mathsf{C}_M^{\rho,c}$ by $\mathsf{C}_M^{\rho,c} = \langle M\{\rho\}, c, \emptyset, \emptyset, \nu(\rho) \cup \{c\}, \emptyset\rangle$.*

Note that traces from $\mathbf{Tr}_{\text{HOSC}}(\mathsf{C}_M^{\rho,c})$ will be $(\nu(\rho) \cup \{c\}, \emptyset)$-traces.

Definition 11. *The* HOSC[HOSC] **trace semantics** *of a cr-free HOSC term $\Gamma \vdash M : \tau$ is defined to be*

$$\mathbf{Tr}_{\text{HOSC}}(\Gamma \vdash M : \tau) = \{((\rho, c), t) \mid \rho \text{ is a } \Gamma\text{-assignment}, c : \tau, t \in \mathbf{Tr}_{\text{HOSC}}(\mathsf{C}_M^{\rho,c})\}.$$

Example 5. Recall the term $\vdash M_1^{cwl} : \tau$ from Example 4, the trace $\mathtt{t}_1$ and the configuration $\mathbf{C}$ such that $\mathtt{t}_1 \in \mathbf{Tr}_{\text{HOSC}}(\mathbf{C})$. Because M_1^{cwl} is closed ($\Gamma = \emptyset$), the only Γ-assignment is the empty map $\emptyset$. Thus, $\mathbf{C} = \mathsf{C}^{\emptyset,c}_{M_1^{cwl}}$, so $((\emptyset, c), \mathtt{t}_1) \in \mathbf{Tr}_{\text{HOSC}}(\vdash M_1^{cwl} : \tau)$.

Having defined active configurations associated to terms, we now turn to defining passive configurations associated to contexts. Let us fix $\diamond \in \text{FNames}_{\text{Unit}\to\text{Unit}}$ and, for each σ, a continuation name $\circ_\sigma \in \text{CNames}_\tau$. Let $\circ = \bigcup_\sigma \{\circ_\sigma\}$. Intuitively, the names $\diamond$ will correspond to $\Downarrow_{err}$ and $\circ_\sigma$ to $\Downarrow_{ter}$.

Recall that $\hat{err}$ stands for $err : \text{Unit} \to \text{Unit}$. Given a heap $h : \Sigma; \hat{err}$, an evaluation context $\Sigma; \hat{err} \vdash K : \tau \to \tau'$ and a substitution $\Sigma; \hat{err} \vdash \gamma : \Gamma$ (as in the definition of $\lesssim_{err}^{\text{HOSC}(ciu)}$), let us replace every occurrence of $\text{cont}_\sigma\, K'$ inside h, K, γ with $\text{cont}_\sigma\, (K', \circ_{\sigma'})$, if K' has type $\sigma \to \sigma'$. Moreover, let us replace every occurrence of the variable err with the function name $\diamond$. This is done to adjust h, K, γ to the extended syntax of the LTS: the upgraded versions are called $h_\circ, \gamma_\circ, K_\circ$.

Next we define the set $\mathbf{AVal}_\Gamma(\gamma)$ of all disjoint decompositions of values from $\gamma_\circ$ into abstract values and the corresponding matchings. Recall that $\Gamma = \{x_1 : \sigma_1, \cdots, x_k : \sigma_k\}$. Below $\vec{A_i}$ stands for $(A_1, \cdots, A_k)$, and $\vec{\gamma_i}$ for $(\gamma_1, \cdots, \gamma_k)$.

$$\begin{aligned}\mathbf{AVal}_\Gamma(\gamma) = \{ \quad & (\vec{A_i}, \vec{\gamma_i}) \mid (A_i, \gamma_i) \in \mathbf{AVal}_{\sigma_i}(\gamma_\circ(x_i)),\ i = 1, \cdots, k; \\ & \nu(A_1), \cdots, \nu(A_k) \text{ mutually disjoint and without } \diamond \quad \}\end{aligned}$$

Definition 12 (Context configuration). *Given Σ, $h : \Sigma; \hat{err}$, $\Sigma; \hat{err} \vdash K : \tau \to \tau'$, $\Sigma; \hat{err} \vdash \gamma : \Gamma$, $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$ and $c : \tau$ ($c \notin \circ$), the corresponding configuration $\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma}$ is defined by*

$$\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma} = \langle \biguplus_{i=1}^{k} \gamma_i \uplus \{c \mapsto K_\circ\}, \{c \mapsto \circ_{\tau'}\}, \biguplus_{i=1}^{k} \nu(A_i) \uplus \{c\} \uplus \circ \uplus \{\diamond\}, h_\circ \rangle.$$

Intuitively, the names $\nu(A_i)$ correspond to calling function values extracted from γ, whereas c corresponds to K. Note that traces in $\mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma})$ will be $(\circ \uplus \{\diamond\}, \biguplus_{i=1}^{k} \nu(A_i) \uplus \{c\})$-traces.

In preparation for the next result, we introduce the following shorthands.

- Given a (N_O, N_P)-trace t, we write $t^\perp$ for the (N_P, N_O)-trace obtained by changing the polarity of each name: $f(A, c')$ becomes $\bar{f}(A, c')$ (and vice versa) and $c(A)$ becomes $\bar{c}(A)$ (and vice versa).
- Given $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$, we define a Γ-assignment $\rho_{\vec{A_i}}$ by $\rho_{\vec{A_i}}(x_i) = A_i$. Note that $\nu(\rho_{\vec{A_i}}) = \biguplus_{i=1}^{k} \mathrm{dom}(\gamma_i)$.

Lemma 4 (Correctness). *Let $\Gamma \vdash M : \tau$ be a cr-free* HOSC *term, let Σ, h, K, γ be as above, $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$, and $c : \tau$ ($c \notin \circ$). Then*

- *$(K[M\{\gamma\}], h) \Downarrow_{err}$ iff there exist t, c' such that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}}, c}_{M})$ and $t^\perp\, \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma})$.*
- *$(K[M\{\gamma\}], h) \Downarrow_{ter}$ iff there exist t, A, σ such that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}}, c}_{M})$ and $t^\perp\, \bar{\circ}_\sigma(A) \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma})$.*

Moreover, t satisfies $\nu(t) \cap (\circ \cup \{\diamond\}) = \emptyset$.

Intuitively, the lemma above confirms that the potential of a term to converge is determined by its traces. Accordingly, we have:

Theorem 1 (Soundness). *For any cr-free* HOSC *terms $\Gamma \vdash M_1, M_2$, if $\mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_2)$ then $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{HOSC}(ciu)} M_2$.*

To prove the converse, we need to know that every odd-length trace generated by a term actually participates in a contextual interaction. This will follow from the lemma below. Note that $\Downarrow_{err}$ relies on even-length traces from the context (Lemma 4).

Lemma 5 (Definability). *Suppose* $\phi \uplus \{\diamond\} \subseteq \mathrm{FNames}$ *and* t *is an even-length* $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. There exists a passive configuration* $\mathbf{C}$ *such that the even-length traces* $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ *are exactly the even-length prefixes of* t *(along with all renamings that preserve types and* $\phi \uplus \{c\} \uplus \circ \uplus \{\diamond\}$*, cf. Remark 2). Moreover,* $\mathbf{C} = \langle \gamma_\circ \cdot [c \mapsto K_\circ], \{c \mapsto \circ_{\tau'}\}, \phi \uplus \{c\} \uplus \circ \uplus \{\diamond\}, h_\circ \rangle$*, where* h, K, γ *are built from* HOSC *syntax.*

Proof (Sketch). The basic idea is to use references in order to record all continuation and function names introduced by the environment. For continuations, the use of $\mathrm{call/cc}_\tau$ is essential. Once stored in the heap, the names can be accessed by terms when needed in P-actions. The availability of throw and references to all O-continuations means that arbitrary answer actions can be scheduled when needed.

Theorem 2 (Completeness). *For any cr-free* HOSC *terms* $\Gamma \vdash M_1, M_2$*,* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{HOSC}(ciu)} M_2$ *implies* $\mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_2)$*.*

Theorems 1, 2 (along with Lemmas 1, 2) imply the following full abstraction results.

Corollary 1 (HOSC Full Abstraction). *Suppose* $\Gamma \vdash M_1, M_2$ *are cr-free* HOSC *terms. Then* $\mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{HOSC}} M_2$ *iff* $\Gamma \vdash M_1 \lesssim_{ter}^{\mathrm{HOSC}} M_2$*.*

Example 6 (Callback with lock [4]). Recall the term $\vdash M_1^{cwl} : ((\mathrm{Unit} \to \mathrm{Unit}) \to \mathrm{Unit}) \times (\mathrm{Unit} \to \mathrm{Int})$ from Example 4, given in Figure 4. We had $\mathtt{t}_1 = \bar{c}(\langle g_1, g_2 \rangle)\ g_1(f_1, c_1)\ \bar{f}_1((), c_2)\ c_2(())\ \bar{c_1}(())\ c_2(())\ \bar{c_1}(())\ g_2((), c_3)\ \bar{c_3}(2) \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_1^{cwl}})$.

Define $\mathtt{t}_2$ to be $\mathtt{t}_1$ except that its last action $\bar{c_3}(2)$ is replaced with $\bar{c_3}(1)$. Observe that $\mathtt{t}_1 \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_1^{cwl}}) \setminus \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_2^{cwl}})$ and $\mathtt{t}_2 \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_2^{cwl}}) \setminus \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_1^{cwl}})$, i.e. by the Corollary above the terms are incomparable wrt $\lesssim_{err}^{\mathrm{HOSC}}$. However, they are equivalent wrt $\lesssim_{err}^{\mathbf{x}}$ for $\mathbf{x} \in \{\mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$ [8].

The above Corollary also provides a handle to reason about equivalence via trace equivalence. Sometimes this can be done directly on the LTS, especially when γ can be kept bounded.

Example 7 (Counter [28]). For $i \in \{1, 2\}$, consider the terms $\vdash M_i : (\mathrm{Unit} \to \mathrm{Unit}) \times (\mathrm{Unit} \to \mathrm{Int})$ given by $M_i \equiv \mathrm{let}\, x = \mathrm{ref}\, 0 \,\mathrm{in}\, \langle \mathsf{inc}_i, \mathsf{get}_i \rangle$, where $\mathsf{inc}_1 \equiv (\lambda y. x := !x + 1)$, $\mathsf{inc}_2 \equiv (\lambda y. x := !x - 1)$, $\mathsf{get}_1 \equiv \lambda z.!x$, $\mathsf{get}_2 \equiv \lambda z.-!x$. In this case, $\mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\emptyset,c}_{M_i})$ contains (prefixes of) traces of the form $\bar{c}(\langle g, h \rangle)\, t$, where t is built from segments of two kinds: either $g((), c_i)\ \bar{c_i}(())$ or $h((), c_i')\ \bar{c}_i'(n)$, where the c_is and c_i's are pairwise different. Moreover, in the latter case, n must be equal to the number of preceding actions of the form $g((), c_i)$. For this example, trace equality could be established by induction on the length of trace. Consequently, $M_1 \cong_{err}^{\mathrm{HOSC}} M_2$.

4 GOSC[HOSC]

Recall that GOSC is the fragment of HOSC in which general storage is restricted to values of *ground* type, i.e. arithmetic/boolean constants, the associated reference names, references to those names and so on. In what follows, we are going to provide characterizations of $\lesssim_{err}^{\mathrm{GOSC}}$ via trace inclusion. Recall that, by Lemma 2, $\lesssim_{err}^{\mathrm{GOSC}} = \lesssim_{ter}^{\mathrm{GOSC}}$. Note that we work in an asymmetric setting with terms belonging to HOSC being more powerful than contexts.

We start off by identifying several technical consequences of the restriction to GOSC syntax. First we observe that GOSC internal reductions never contribute extra names.

Lemma 6. *Suppose* $(M,c,h) \to (M',c',h')$*, where* M *is a* GOSC *term and* h *is a* GOSC *heap. Then* $\nu(M)\cup\{c\} \supseteq \nu(M')\cup\{c'\}$*.*

Proof. By case analysis. All defining rules for $\to$, with the exception of the $(K[!\ell],h) \to (K[h(\ell)],h)$ rule, are easily seen to satisfy the Lemma (no function or continuation names are added). However, if the heap is restricted to storing elements of type ι (as in GOSC) then $h(\ell)$ will never contain a name, so the Lemma follows.

The lemma has interesting consequences for the shape of traces generated by the context configurations $\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i},c}$ if they are built from GOSC syntax. Recall that P-actions have the form $\bar{f}(A,c')$ or $\bar{c}(A)$, where f,c are names introduced by O. It turns out that when h,K,γ are restricted to GOSC, more can be said about the origin of the names in traces generated by $\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i},c}$: they will turn out to come from a restricted set of names introduced by O, which we identify below. The definition below is based on following the justification structure of a trace – recall that one action is said to justify another if the former introduces a name that is used for communication in the latter.

Definition 13. *Suppose* $\phi \uplus \{\diamond\} \subseteq \mathrm{FNames}$ *and* $c \in \mathrm{CNames}$*. Let* t *be an odd-length* $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. The set* $\mathsf{Vis}_P(t)$ *of P-visible names of* t *is defined as follows.*

$$
\begin{array}{rcll}
\mathsf{Vis}_P(t\ \ c'(A')) &=& \{\diamond\}\cup\circ\cup\nu(A') & c'=c\\
\mathsf{Vis}_P(t\ \ \bar{f''}(A'',c')\ \ t'\ \ c'(A')) &=& \mathsf{Vis}_P(t)\cup\nu(A') & c'\neq c\\
\mathsf{Vis}_P(t\ \ f'(A',c')) &=& \{\diamond\}\cup\circ\cup\nu(A')\cup\{c'\} & f'\in\phi\\
\mathsf{Vis}_P(t\ \ \bar{f''}(A'',c'')\ \ t'\ \ f'(A',c')) &=& \mathsf{Vis}_P(t)\cup\nu(A')\cup\{c'\} & f'\in\nu(A'')\\
\mathsf{Vis}_P(t\ \ \bar{c''}(A'')\ \ t'\ \ f'(A',c')) &=& \mathsf{Vis}_P(t)\cup\nu(A')\cup\{c'\} & f'\in\nu(A'')
\end{array}
$$

Note that, in the inductive cases, the definition follows links between names introduced by P and the point of their introduction, names introduced in-between are ignored. Here readers familiar with game semantics will notice similarity to the notion of P-view [12].

Next we specify a property of traces that will turn out to be satisfied by configurations corresponding to GOSC contexts.

Definition 14. *Suppose $\phi \uplus \{\diamond\} \subseteq \text{FNames}$ and $c \in \text{CNames}$. Let t be a $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$-trace starting with an O-action. t is called* **P-visible** *if*

- *for any even-length prefix $t'\,\bar{f}(A,c)$ of t, we have $f \in \mathsf{Vis}_P(t')$,*
- *for any even-length prefix $t'\,\bar{c}(A)$ of t, we have $c \in \mathsf{Vis}_P(t')$.*

Lemma 7. *Consider $\mathbf{C} = \mathsf{C}^{\vec{\gamma_i};c}_{h,K,\gamma}$, where h, K, γ are from GOSC and $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$. Then all traces in $\mathbf{Tr}_{\text{HOSC}}(\mathbf{C})$ are P-visible.*

The Lemma above shows that contextual interactions with GOSC contexts rely on restricted traces. We shall now modify the HOSC[HOSC] LTS to capture the restriction. Note that, from the perspective of the term, the above constraint is a constraint on the use of names by O (context), so we need to talk about O-available names instead. This dual notion is defined below.

Definition 15. *Suppose $\phi \subseteq \text{FNames}$ and $c \in \text{CNames}$. Let t be a $(\phi \uplus \{c\}, \emptyset)$-trace of odd length. The set $\mathsf{Vis}_O(t)$ of* **O-visible names** *of t is defined as follows.*

$$
\begin{array}{rll}
\mathsf{Vis}_O(t\ \ \bar{c'}(A')) = \nu(A') & & c' = c \\
\mathsf{Vis}_O(t\ \ f''(A'',c')\ \ t'\ \ \bar{c'}(A')) = \mathsf{Vis}_O(t) \cup \nu(A') & & c' \neq c \\
\mathsf{Vis}_O(t\ \ \bar{f'}(A',c')) = \nu(A') \cup \{c'\} & & f' \in \phi \\
\mathsf{Vis}_O(t\ \ f''(A'',c'')\ \ t'\ \ \bar{f'}(A',c')) = \mathsf{Vis}_O(t) \cup \nu(A') \cup \{c'\} & & f' \in \nu(A'') \\
\mathsf{Vis}_O(t\ \ c''(A'')\ \ t'\ \ \bar{f'}(A',c')) = \mathsf{Vis}_O(t) \cup \nu(A') \cup \{c'\} & & f' \in \nu(A'')
\end{array}
$$

Analogously, a $(\phi \uplus \{c\}, \emptyset)$-trace t is **O-visible** *if, for any even-length prefix $t'\,f(A,c)$ of t, we have $f \in \mathsf{Vis}_O(t')$ and, for any even-length prefix $t'\,c(A)$ of t, we have $c \in \mathsf{Vis}_O(t')$.*

Example 8. Recall the trace

$$\mathtt{t}_1 = \bar{c}(\langle g_1, g_2\rangle)\ \ g_1(f_1, c_1)\ \ \bar{f_1}((), c_2)\ \ c_2(())\ \ \bar{c_1}(())\ \ c_2(())\ \ \bar{c_1}(())\ \ g_2((), c_3)\ \ \bar{c_3}(2)$$

from previous examples. Observe that

$$
\begin{array}{r}
\mathsf{Vis}_O(\bar{c}(\langle g_1, g_2\rangle)\ g_1(f_1, c_1)\ \bar{f_1}((), c_2)) = \{g_1, g_2, c_2\} \\
\mathsf{Vis}_O(\bar{c}(\langle g_1, g_2\rangle)\ g_1(f_1, c_1)\ \bar{f_1}((), c_2)\ c_2(())\ \bar{c_1}(())) = \{g_1, g_2\}
\end{array}
$$

Consequently, the first use of $c_2(())$ in $\mathtt{t}_1$ does not violate O-visibility, but the second one does.

In Figure 6, we present a new LTS, called the GOSC[HOSC] LTS, which will turn out to capture $\lesssim^{\text{GOSC}}_{err}$ through trace inclusion. It is obtained from the HOSC[HOSC] LTS by restricting O-actions to those that rely on O-visible names. Technically, this is done by enriching configurations with an additional component $\mathcal{F}$, which maintains historical information about O-available names immediately before each O-action. After each P-action, $\mathcal{F}$ is accessed to calculate the current set $\mathcal{V}$ of O-available names according to the definition of O-availability and only O-actions compatible with O-availability are allowed to proceed (due

$$
\begin{array}{l|l}
(P\tau) & \langle M, c, \gamma, \xi, \phi, h, \mathcal{F}\rangle \xrightarrow{\tau} \langle N, c', \gamma, \xi, \phi, h', \mathcal{F}\rangle \\
& \text{when } (M, c, h) \to (N, c', h') \\
(PA) & \langle V, c, \gamma, \xi, \phi, h, \mathcal{F}\rangle \xrightarrow{\bar{c}(A)} \langle \gamma \cdot \gamma', \xi, \phi \uplus \nu(A), h, \mathcal{F}, \mathcal{F}(c) \uplus \nu(A)\rangle \\
& \text{when } c : \sigma \text{ and } (A, \gamma') \in \mathbf{AVal}_\sigma(V) \\
(PQ) & \langle K[fV], c, \gamma, \xi, \phi, h, \mathcal{F}\rangle \xrightarrow{\bar{f}(A,c')} \langle \gamma \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \phi \uplus \phi', h, \mathcal{F}, \mathcal{F}(f) \uplus \phi'\rangle \\
& \text{when } f : \sigma \to \sigma',\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ c' : \sigma' \text{ and } \phi' = \nu(A) \uplus \{c'\} \\
(OA) & \langle \gamma, \xi, \phi, h, \mathcal{F}, \mathcal{V}\rangle \xrightarrow{c(A)} \langle K[A], c', \gamma, \xi, \phi \uplus \nu(A), h, \mathcal{F} \cdot [\nu(A) \mapsto \mathcal{V}]\rangle \\
& \text{when } c \in \mathcal{V},\ c : \sigma,\ A : \sigma,\ \gamma(c) = K,\ \xi(c) = c' \\
(OQ) & \langle \gamma, \xi, \phi, h, \mathcal{F}, \mathcal{V}\rangle \xrightarrow{f(A,c)} \langle VA, c, \gamma, \xi, \phi \uplus \phi', h, \mathcal{F} \cdot [\phi' \mapsto \mathcal{V}]\rangle \\
& \text{when } f \in \mathcal{V},\ f : \sigma \to \sigma',\ A : \sigma,\ c : \sigma',\ \gamma(f) = V \text{ and } \phi' = \nu(A) \uplus \{c\}
\end{array}
$$

Given $N \subseteq$ Names, $[N \mapsto \mathcal{V}]$ stands for the map $[n \mapsto \mathcal{V} \mid n \in N]$.

Fig. 6. GOSC[HOSC] LTS

to the $f \in \mathcal{V}$, $c \in \mathcal{V}$ side conditions). We write $\mathbf{Tr}_{\mathrm{GOSC}}(\mathbf{C})$ for the set of traces generated from $\mathbf{C}$ in the GOSC[HOSC] LTS.

Recall that, given a Γ-assignment ρ, term $\Gamma \vdash M : \tau$ and $c \in \mathrm{CNames}_\tau$, the active configuration $\mathsf{C}_M^{\rho,c}$ was defined by $\mathsf{C}_M^{\rho,c} = \langle M\{\rho\}, c, \emptyset, \emptyset, \nu(\rho) \cup \{c\}, \emptyset\rangle$. We need to upgrade it to the LTS by initializing the new component to the empty map: $\mathsf{C}_{M,vis}^{\rho,c} = \langle M\{\rho\}, c, \emptyset, \emptyset, \nu(\rho) \cup \{c\}, \emptyset, \emptyset\rangle$.

Definition 16. *The* GOSC[HOSC] **trace semantics** *of a cr-free HOSC term* $\Gamma \vdash M : \tau$ *is defined to be*

$$\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M : \tau) = \{((\rho, c), t) \mid \rho \text{ is a } \Gamma\text{-assignment}, c : \tau, t \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M,vis}^{\rho,c})\}.$$

By construction, it follows that

Lemma 8. $t \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M,vis}^{\rho,c})$ *iff* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_M^{\rho,c})$ *and* t *is O-visible.*

Noting that the witness trace t from Lemma 4 is O-visible iff $t^\perp\, \bar{\diamond}((), c')$ is P-visible, we can conclude that, for GOSC, the traces relevant to $\Downarrow_{err}$ are O-visible, which yields:

Theorem 3 (Soundness). *For any cr-free* HOSC *terms* $\Gamma \vdash M_1,\ M_2$*, if* $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$ *then* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}(ciu)} M_2$.

To prove the converse, we need a new definability result. This time we are only allowed to use GOSC syntax, but the target is also more modest: we are only aiming to capture P-visible traces.

Lemma 9 (Definability). *Suppose* $\phi \uplus \{\diamond\} \subseteq$ FNames *and* t *is an even-length* P-visible $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. There exists a passive configuration* $\mathbf{C}$ *such that the even-length traces in* $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ *are exactly the even-length prefixes of* t *(along with all renamings that preserve types and* $\phi \uplus \{c\} \uplus \circ \uplus \{\diamond\}$*). Moreover,* $\mathbf{C} = \langle \gamma_\circ \cdot [c \mapsto K_\circ], \{c \mapsto \circ_{\tau'}\}, \phi \uplus \{c\} \uplus \circ \uplus \{\diamond\}, h_\circ\rangle$*, where* h, K, γ *are built from* GOSC *syntax.*

Proof (Sketch). This time we cannot rely on references to recall on demand all continuation and function names introduced by the environment. However, because t is P-visible, it turns the uses of the names can be captured through variable bindings ($\lambda x.\cdots$ for function and $\text{call/cc}_\tau(x.\dots)$ for continuation names). Using throw, we can then force an arbitrary answer action, as long as it uses a P-available name. To select the right action at each step, we branch on the value of a single global reference of type ref Int that keeps track of the number of steps simulated so far.

Completeness now follows because, for a potential O-visible witness t from Lemma 4, one can create a corresponding context by invoking the Definability result for $t^{\perp}\,\bar{\diamond}((), c')$. It is crucial that the addition of $\bar{\diamond}((), c')$ does not break P-visibility ($\diamond$ is P-visible).

Theorem 4 (Completeness). *For any cr-free* HOSC *terms* $\Gamma \vdash M_1, M_2$, *if* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}(ciu)} M_2$ *then* $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$.

Altogether, Theorems 3, 4 (along with Lemma 1) imply the following result.

Corollary 2 (GOSC **Full Abstraction).** *Suppose* $\Gamma \vdash M_1, M_2$ *are cr-free* HOSC *terms. Then* $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}(ciu)} M_2$ *iff* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}} M_2$.

Example 9. In the *Callback with lock* example (Example 6), we exhibited traces $\mathtt{t}_1, \mathtt{t}_2$ that separated M_1^{cwl}, M_2^{cwl} wrt $\lesssim_{err}^{\mathrm{HOSC}}$. Example 8 shows that neither trace is O-visible, i.e. they cannot be found in $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1)$ or $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$. Thus, the two traces cannot be used to separate M_1^{cwl}, M_2^{cwl} wrt $\lesssim_{err}^{\mathrm{GOSC}}$. As already mentioned, this is in fact impossible: we have $\vdash M_1^{cwl} \cong_{err}^{\mathrm{GOSC}} M_2^{cwl}$.

Example 10 (Well-bracketed state change [4]). Consider the following two terms

$$M_1^{wbsc} \triangleq \text{let}\, x = \text{ref}\, 0 \,\text{in}\, \lambda f.(x := 0; f(); x := 1; f(); !x)$$
$$M_2^{wbsc} \triangleq \lambda f.(f(); f(); 1).$$

of type $\tau = (\text{Unit} \to \text{Unit}) \to \text{Int}$, let

$$\mathtt{t}_3 = \bar{c}(g) \;\; g(f_1, c_1) \;\; \bar{f}_1((), c_2) \;\; c_2(()) \;\; \bar{f}_1((), c_3) \;\; g(f_2, c_4) \;\; \bar{f}_2((), c_5) \;\; c_3(()) \;\; \bar{c}_1(0)$$

and let $\mathtt{t}_4$ be obtained from $\mathtt{t}_3$ by changing 0 in the last action to 1. One can check that both traces are O-visible: in particular, the action $c_3(())$ is not a violation because

$$\mathsf{Vis}_O(\bar{c}(g)\; g(f_1, c_1)\; \bar{f}_1((), c_2)\; c_2(())\; \bar{f}_1((), c_3)\; g(f_2, c_4)\; \bar{f}_2((), c_5)) = \{g, c_3, c_5\}.$$

Moreover, we have $\mathtt{t}_3 \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}^{\emptyset,c}_{M_1^{wbsc}}) \backslash \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}^{\emptyset,c}_{M_2^{wbsc}})$ and $\mathtt{t}_4 \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}^{\emptyset,c}_{M_2^{wbsc}}) \backslash \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}^{\emptyset,c}_{M_1^{wbsc}})$. By the Corollary above, we can conclude that M_1^{wbsc}, M_2^{wbsc} are incomparable wrt $\lesssim_{err}^{\mathrm{GOSC}}$. However, they turn out to be $\cong_{err}^{\mathrm{HOS}}$- and $\cong_{err}^{\mathrm{GOS}}$-equivalent.

5 HOS[HOSC]

Recall that HOS is the fragment of HOSC that does not feature continuation types and the associated syntax. In what follows we are going to provide alternative characterisations of $\lesssim_{err}^{\mathrm{HOS}}$ and $\lesssim_{ter}^{\mathrm{HOS}}$ in terms of trace inclusion and complete trace inclusion respectively.

We start off by identifying several technical consequences of the restriction to HOS syntax. First we observe that HOS internal reductions never change the associated continuation name.

Lemma 10. *If $(M, c, h) \to (M', c', h')$, M is a* HOS *term and h is a* HOS *heap then $c = c'$.*

Proof. The only rule that could change c is the rule for throw, but it is not part of HOS.

The lemma has a bearing on the shape of traces generated by the (passive) configurations $\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i},c}$ corresponding to HOS contexts. In the presence of throw and storage for continuations, it was possible for P to play answers involving arbitrary continuation names introduced by O. By Lemma 10, in HOS this will be restricted to the continuation name of the current configuration, which will restrict the shape of possible traces. Below we identify the continuation name $top_P(t)$ that becomes the relevant name after trace t. If the last move was an O-question then the continuation name introduced by that move will become that name. Otherwise, we track a chain of answers and questions, similarly to the definition of P-visibility.

Observe that, because h, K, γ are from HOS, $\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i},c}$ will generate $(\{\circ_{\tau'}, \diamond\}, \phi \uplus \{c\})$-traces, where τ' is the result type of K, because $h_\circ = h, K_\circ = K, \gamma_\circ = \gamma$.

Definition 17. *Suppose $\phi \uplus \{\diamond\} \subseteq$ FNames and $c \in$ CNames. Let t be a $(\{\circ_{\tau'}, \diamond\}, \phi \uplus \{c\})$-trace of odd length starting with an O-action. The continuation name $top_P(t)$ is defined as follows.*

$$\begin{aligned} top_P(t\ c(A)) &= \circ_{\tau'} \\ top_P(t_1\ \bar{f}(A'', c')\ t_2\ c'(A')) &= top_P(t_1) \\ top_P(t\ f(A', c')) &= c' \end{aligned}$$

We say that a $(\{\circ_{\tau'} \cup \{\diamond\}, \phi \uplus \{c\})$-trace t starting with an O-action is **P-bracketed** *if, for any prefix $t'\ \bar{c'}(A)$ of t (i.e. any prefix ending with a P-answer), we have $c' = top_P(t')$.*

Lemma 11. *Consider $\mathbf{C} = \mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i},c}$, where h, K, γ are from* HOS *and $(\vec{A_i}, \vec{\gamma_i}) \in$ $\mathbf{AVal}_{\Gamma}(\gamma)$. Then all traces in $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ are P-bracketed.*

The Lemma above characterizes the restrictive nature of contextual interactions with HOS contexts. Next we shall constrain the HOSC[HOSC] LTS accordingly to capture the restriction. Note that, from the point of view of the term, the above-mentioned constraint concerns the use of continuation names by O (the context), so we need to talk about O-bracketing instead. This dual notion of "a top name for O" is specified below.

$$
\begin{array}{l|lcl}
(P\tau) & \langle M, c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\tau} & \langle N, c', \gamma, \xi, \phi, h'\rangle \\
& \multicolumn{3}{l}{\text{when } (M, c, h) \to (N, c', h')} \\
(PA) & \langle V, c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\bar{c}(A)} & \langle \gamma \cdot \gamma', \xi, \phi \uplus \nu(A), h, c'\rangle \\
& \multicolumn{3}{l}{\text{when } c : \sigma,\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ \xi(c) = c'} \\
(PQ) & \langle K[fV], c, \gamma, \xi, \phi, h\rangle & \xrightarrow{\bar{f}(A,c')} & \langle \gamma \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \phi \uplus \nu(A) \uplus \{c'\}, h, c'\rangle \\
& \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma',\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ c' : \sigma'} \\
(OA) & \langle \gamma, \xi, \phi, h, c''\rangle & \xrightarrow{c(A)} & \langle K[A], c', \gamma, \xi, \phi \uplus \nu(A), h\rangle \\
& \multicolumn{3}{l}{\text{when } c = c'',\ c : \sigma,\ A : \sigma,\ \gamma(c) = K,\ \xi(c) = c'} \\
(OQ) & \langle \gamma, \xi, \phi, h, c''\rangle & \xrightarrow{f(A,c)} & \langle VA, c, \gamma, \xi \cdot [c \mapsto c''], \phi \uplus \nu(A) \uplus \{c\}, h\rangle \\
& \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma',\ A : \sigma,\ c : \sigma',\ \gamma(f) = V}
\end{array}
$$

Fig. 7. HOS[HOSC] LTS

Definition 18. *Suppose* $\phi \subseteq \mathrm{FNames}$ *and* $c \in \mathrm{CNames}$. *Let* t *be a* $(\phi \uplus \{c\}, \emptyset)$*-trace of odd length. The continuation name* $top_O(t)$ *is defined as follows. In the first case, the value is* $\bot$ *(representing "none"), because* c *is the top continuation passed by the environment to the term (if it gets answered there is nothing left to answer).*

$$
\begin{aligned}
top_O(t\ \bar{c}(A)) &= \bot \\
top_O(t_1\ f(A'', c')\ t_2\ \bar{c}'(A')) &= top_O(t_1) \\
top_O(t\ \bar{f}(A', c')) &= c'
\end{aligned}
$$

We say that a $(\phi \uplus \{c\}, \emptyset)$*-trace* t *is* **O-bracketed** *if, for any prefix* $t'\ \bar{c}'(A)$ *of* t *(i.e. any prefix ending with a P-answer), we have* $c' = top_O(t')$.

In Figure 7, we present a new LTS, called the HOS[HOSC] LTS, which will turn out to capture $\lesssim^{\mathrm{HOS}}_{err}$. It is obtained from the HOSC[HOSC] LTS by restricting O-actions to those that satisfy O-bracketing. Technically, this is done by enriching passive configurations with a component for storing the current value of $top_O(t)$. In order to maintain this information, we need to know which continuation will become the top one if P plays an answer. This can be done with a map that maps continuations introduced by O to other continuations. Because its flavour is similar to ξ (which is a map from continuations introduced by P) we integrate this information into ξ. The $c = c''$ side condition then enforces O-bracketing. We shall write $\mathbf{Tr}_{\mathrm{HOS}}(\mathbf{C})$ for the set of traces generated from $\mathbf{C}$ in the HOS[HOSC] LTS.

Recall that, given a Γ-assignment ρ, term $\Gamma \vdash M : \tau$ and $c : \tau$, the active configuration $\mathsf{C}^{\rho;c}_M$ was defined by $\mathsf{C}^{\rho;c}_M = \langle M\{\rho\}, c, \emptyset, \emptyset, \nu(\rho) \cup \{c\}, \emptyset\rangle$. We upgrade it to the new LTS by setting $\mathsf{C}^{\rho;c}_{M,bra} = \langle M\{\rho\}, c, \emptyset, [c \mapsto \bot], \nu(\rho) \cup \{c\}, \emptyset, \emptyset\rangle$. This initializes ξ in such a way that, after $\bar{c}(A)$ is played, the extra component will be set to $\bot$, where $\bot$ is a special element not in CNames.

Definition 19. *The* HOS[HOSC] **trace semantics** *of a cr-free HOSC term* $\Gamma \vdash M : \tau$ *is defined to be*

$$\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M : \tau) = \{((\rho, c), t) \mid \rho \textit{ is a } \Gamma\textit{-assignment}, \ c : \tau, \ t \in \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M,bra})\}.$$

By construction, it follows that

Lemma 12. $t \in \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M,bra})$ *iff* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho,c}_{M})$ *and* t *is O-bracketed.*

Noting that the witness trace t from Lemma 4 is O-bracketed iff $t^{\perp}\,\bar{\diamond}((), c')$ is P-bracketed, we can conclude that, for HOS, the traces relevant to $\Downarrow_{err}$ are O-bracketed, which yields:

Theorem 5 (Soundness). *For any cr-free* HOSC *terms* $\Gamma \vdash M_1, M_2$*, if* $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ *then* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}(ciu)}_{err} M_2$.

For the converse, we establish another definability result, this time for a P-bracketed trace.

Lemma 13 (Definability). *Suppose* $\phi \uplus \{\diamond\} \subseteq$ FNames *and* t *is an even-length* P-bracketed $(\{\circ_{\tau'}, \diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. There exists a passive configuration* $\mathbf{C}$ *such that the even-length traces* $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ *are exactly the even-length prefixes of* t *(along with all renamings that preserve types and* $\phi \uplus \{c, \circ_{\tau'}, \diamond\}$*). Moreover,* $\mathbf{C} = \langle \gamma \cdot [c \mapsto K], \{c \mapsto \circ_{\tau'}\}, \phi \uplus \{c, \circ_{\tau'}, \diamond\}, h\rangle$*, where* h, K, γ *are built from* HOS *syntax.*

Proof (Sketch). Our argument for HOSC is structured in such a way that, for a P-bracketed trace, there is no need for continuations (throwing and continuation capture are not necessary).

Completeness now follows because, for a potential witness trace t from Lemma 4, one can create a corresponding context by invoking the Definability result for $t^{\perp}\,\bar{\diamond}((), c')$. It is crucial that the addition of $\bar{\diamond}((), c')$ does not break P-bracketing (it does not, because the action is a question).

Theorem 6 (Completeness). *For any cr-free* HOSC *terms* $\Gamma \vdash M_1, M_2$*, if* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}(ciu)}_{err} M_2$ *then* $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$.

Altogether, Theorems 5, 6 (along with Lemma 1) imply the following result.

Corollary 3 (HOS **Full Abstraction).** *Suppose* $\Gamma \vdash M_1, M_2$ *are cr-free* HOSC *terms. Then* $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}(ciu)}_{err} M_2$ *iff* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}}_{err} M_2$.

Example 11 (Assignment/callback commutation [27]). For $i \in \{1, 2\}$, let $f : \mathrm{Unit} \to \mathrm{Unit} \vdash M_i : \mathrm{Unit} \to \mathrm{Unit}$ be defined by:

$$M_1 \triangleq \mathrm{let}\, n = \mathrm{ref}\,(0)\, \mathrm{in}\, \lambda y^{\mathrm{Unit}}.\mathrm{if}\ (!n > 0)\ ()\ (n := 1; f()),$$
$$M_2 \triangleq \mathrm{let}\, n = \mathrm{ref}\,(0)\, \mathrm{in}\, \lambda y^{\mathrm{Unit}}.\mathrm{if}\ (!n > 0)\ ()\ (f(); n := 1).$$

Operationally, one can see that $f \vdash M_1 \not\lesssim^{\mathrm{HOS}}_{err} M_2$ due to the following HOS context: let $r = \mathrm{ref}\,(\lambda y.y)$ in (let $f = \lambda y.(!r)()$ in $(r := \bullet; (!r)()))$; err. In our framework, this is confirmed by the trace

$$\mathtt{t}_5 \quad = \quad \bar{c}(g) \quad g((), c_1) \quad \bar{f}((), c_2) \quad g((), c_2) \quad \bar{c_2}(()),$$

which is in $\mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_1}) \setminus \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_2})$. On the other hand,

$$\mathtt{t}_6 \quad = \quad \bar{c}(g) \quad g((), c_1) \quad \bar{f}((), c_2) \quad g((), c_2) \quad \bar{f}((), c_3)$$

is in $\mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_2}) \setminus \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_1})$, so the terms are incomparable. Note, however, that both traces break O-visibility: specifically, we have

$$\mathsf{Vis}_O(\bar{c}(g)\ g((), c_1)\ \bar{f}((), c_2)) = \{c_2\},$$

so the $g((), c_2)$ action violates the condition. Consequently, the traces do not preclude $f \vdash M_1 \cong^{\mathbf{x}}_{err} M_2$ for $\mathbf{x} \in \{\mathrm{GOSC}, \mathrm{GOS}\}$.

For $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}\}$, $\lesssim^{\mathbf{x}}_{err}$ and $\lesssim^{\mathbf{x}}_{ter}$ coincide. Intuitively, this is because the presence of continuations in the context makes it possible to make an escape at any point. In contrast, for HOS, the context must run to completion in order to terminate.

At the technical level, one can appreciate the difference when trying to transfer our results for $\lesssim^{\mathrm{HOS}(ciu)}_{err}$ to $\lesssim^{\mathrm{HOS}(ciu)}_{ter}$. Recall that, according to Lemma 4, $\Downarrow_{ter}$ relies on a witness trace t such that the context configuration generates $t^{\perp}\ \circ^{-}_{\tau'}()$. In HOS, the latter must satisfy P-bracketing, so we need $top_P(t^{\perp}) = \circ_{\tau'}$. Note that this is equivalent to $top_O(t) = \perp$. Consequently, only such traces are relevant to observing $\Downarrow_{ter}$.

Let us call an odd-length O-bracketed $(\phi \uplus \{c\}, \emptyset)$-trace t ***complete*** if $top_O(t) = \perp$. Let us write $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq_c \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ if we have $((\rho, c), t) \in \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ whenever $((\rho, c), t) \in \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1)$ and t is complete. Following our methodology, one can then show:

Theorem 7 **(HOS Full Abstraction for $\lesssim^{\mathrm{HOS}}_{ter}$).** *Suppose $\Gamma \vdash M_1, M_2$ are cr-free* HOSC *terms. Then* $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq_c \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}(ciu)}_{ter} M_2$ *iff* $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}}_{ter} M_2$.

Example 12. Let $M_1 \equiv \lambda f^{\mathrm{Unit}\to\mathrm{Unit}}.f(); \Omega_{\mathrm{Unit}}$ and $M_2 \equiv \lambda f^{\mathrm{Unit}\to\mathrm{Unit}}.\Omega_{\mathrm{Unit}}$. We will see that $\vdash M_1 \not\lesssim^{\mathrm{HOS}}_{err} M_2$ but $\vdash M_1 \lesssim^{\mathrm{HOS}}_{ter} M_2$. To see this, note that $\mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_1})$ contains prefixes of $\bar{c}(g)\ g(f, c_1)\ \bar{f}((), c_2)\ c_2(())$, while $\mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho,c}_{M_2})$ only those of $\bar{c}(g)\ g(f, c_1)$. Observe that the only complete trace among them is $\bar{c}(g)$. The trace $t = \bar{c}(g)\ g(f, c_1)\ \bar{f}((), c_2)$ is not complete, because $top_O(t) = c_2$. Consequently, $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \not\subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$ but $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq_c \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$.

The theorem above generalizes the characterisation of contextual equivalence between HOS terms with respect to HOS contexts [23], where trace completeness means both O- and P-bracketing and "all questions must be answered". Our definition of completeness is weaker (O-bracketing + "the top question must be answered"), because it also covers HOSC terms. However, in the presence of both O- and P-bracketing, i.e. for HOS terms, they will coincide.

6 GOS[HOSC]

Recall that GOS features ground state only and, technically, is the intersection of GOSC and HOS. Consequently, it follows from the previous sections that GOS contexts yield configurations that satisfy both P-visibility and P-bracketing. For such traces, the definability result for GOSC yields a GOS context. Thus, in a similar fashion to the previous sections, we can conclude that O-visible and O-bracketed traces underpin $\lesssim_{err}^{\mathrm{GOS}}$. To define the GOS LTS we simply combine the restrictions imposed in the previous sections, and define $\mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M)$ analogously. We present the LTS in Appendix F. The results on $\lesssim_{ter}^{\mathrm{GOS}}$ from the previous section also carry over to GOS.

Theorem 8 (GOS **Full Abstraction).** *Suppose* $\Gamma \vdash M_1, M_2$ *are cr-free* HOSC *terms. Then:*

- $\mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOS}(ciu)} M_2$ *iff* $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOS}} M_2$.
- $\mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M_1) \subseteq_c \mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M_2)$ *iff* $\Gamma \vdash M_1 \lesssim_{ter}^{\mathrm{GOS}(ciu)} M_2$ *iff* $\Gamma \vdash M_1 \lesssim_{ter}^{\mathrm{GOS}} M_2$.

7 Concluding remarks

Asymmetry Our framework is able to deal with asymmetric scenarios, where programs are taken from HOSC, but are tested with contexts from weaker fragments. For example, we can compare the following two HOSC programs, where $f : ((\mathrm{Unit} \to \mathrm{Unit}) \to \mathrm{Unit}) \to \mathrm{Unit}$ is a free identifier.

let b = ref **ff** in callcc(y.
    f(λg.b := **tt**; g(); throw() to y);
    if !b then () else div)
	callcc(y.
	    f(λg.g(); throw() to y);
	    div)

with div representing divergence. The terms happen to be $\cong_{err}^{\mathrm{HOS}}$-equivalent, but not $\cong_{err}^{\mathrm{HOSC}}$-equivalent.

To see this at the intuitive level, we make the following observations.

- Firstly, we observe that, to distinguish the terms, f should use its argument. Otherwise, the value of b will remain equal to **ff**, and the only subterm that distinguishes the terms ('if !b then () else div') will play the same role as div in the second term.
- Secondly, if f does use its argument, then b will be set to **tt** in the first program, raising the possibility of distinguishing the terms. However, if we allow HOS contexts only then, since the argument to f was used, it will have to run to completion, before 'if !b then () else div' is reached. Consequently, we will encounter 'throw () to y' earlier and never reach 'if !b then () else div)'. This is represented by the trace

$$\bar{f}(h, c_1) \qquad h(g, c_2) \qquad \bar{g}((), c_3) \qquad c_3(()) \qquad \bar{c}(())$$

This trace is O-bracketed, but not P-bracketed since Player uses throw to answer directly to the initial continuation c rather than c_2.
– Finally, if HOSC contexts are allowed, it is possible to reach 'if !b then () else div)' b set to **tt**. This is represented by the trace

$$\bar{f}(h, c_1) \qquad h(g, c_2) \qquad \bar{g}((), c_3) \qquad c_1(()) \qquad \bar{c}(())$$

This trace is not O-bracketed, because c_1 is answered rather than c_3, like above. Consequently, the trace witnesses termination of the first term, but the second term would diverge during interaction with the same context.

We plan to explore the opportunities presented by this setting in the future, especially with respect to fully abstract translations, for example, from HOSC to GOS.

Richer Types Recall that our full abstraction results are stated for cr-free terms, terms with cont- and ref-free types at the boundary. Here we first discuss how to extend them to more complicated types.

To deal with reference type at the boundary, i.e. location exchange, one needs to generalize the notion of traces, so that they can carry, for each action, a heap representing the values stored in the disclosed part of the heap, as in [23, 27]. The extension to sum, recursive and empty types seems conceptually straightforward, by simply extending the definition of abstract values for these types, following the similar notion of ultimate pattern in [24]. The same idea should apply to allow continuation types at the boundary. Operational game semantics for an extension of HOS with polymorphism has been explored in [15].

Innocence On the other hand, all of the languages we considered were stateful. In the presence of state, all of the actions that are represented by labels (and their order and frequency) can be observed, because they could generate a side-effect. A natural question to ask whether the techniques could also be used to provide analogous theorems for purely functional computation, i.e. contexts taken from the language PCF. Here, the situation is different. For example, the terms $f : \mathrm{Int} \to \mathrm{Int} \vdash f(0)$ and $f : \mathrm{Int} \to \mathrm{Int} \vdash \text{if } f(0)\ f(0)\ f(0)$ should be equivalent, even though the sets of their traces are incomparable.

It is known [12] that PCF strategies satisfy a uniformity condition called innocence. Unfortunately, restricting our traces to "O-innocent ones" (like we did with O-visibility and O-bracketing) would not deliver the required characterization. Technically, this is due to the fact that, in our arguments, given a single trace (with suitable properties), we can produce a context that induces the given trace and no other traces (except those implied by the definition of a trace). For innocence, this would not be possible due to the uniformity requirement. It will imply that, although we can find a functional context that generates an innocent trace, it might also generate other traces, which then have to be taken into account when considering contextual testing. This branching property makes it difficult to capture equivalence with respect to functional contexts explicitly, e.g. through traces, which is illustrated by the use of the so-called intrinsic quotient in game models of PCF [2, 12].

8 Related Work

We have presented four operational game models for HOSC, which capture term interaction with contexts built from any of the four sublanguages $\mathbf{x} \in \{$HOSC, GOSC, HOS, GOS$\}$ respectively. The most direct precursor to this work is Laird's trace model for HOS[HOS] [23]. Other frameworks in this spirit include models for objects [18], aspects [16] and system-level code [9]. In [13], Laird's model has been related formally to the denotational game model from [27]. However, in general, it is not yet clear how one can move systematically between the operational and denotational game-based approaches, despite some promising steps reported in [25]. Below we mention other operational techniques for reasoning about contextual equivalence.

In [31], fully abstract Eager-Normal-Form (enf) Bisimulations are presented for an untyped λ-calculus with store and control, similar to HOSC (but with control represented using the $\lambda\mu$-calculus). The bisimulations are parameterised by worlds to model the evolution of store, and bisimulations on contexts are used to deal with control. Like our approach, they are based on symbolic evaluation of open terms. Typed enf-bisimulations, for a language without store and in control-passing style, have been introduced in [24]. Fully-abstract enf-bisimulations are presented in [7] for a language with state only, corresponding to an untyped version of HOS. Earlier works in this strand include [17, 29].

Environmental Bisimulations [19, 30, 32] have also been introduced for languages with store. They work on closed terms, computing the arguments that contexts can provide to terms using an environment similar to our component γ. They have also been extended to languages with call/cc [34] and delimited control operators [5, 6].

Kripke Logical Relations [28, 4, 8] have been introduced for languages with state and control. In [8], a characterization of contextual equivalence for each case $\mathbf{x}[\mathbf{x}]$ ($\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$) is given, using techniques called backtracking and public transitions, which exploit the absence of higher-order store and that of control constructs respectively. Importing these techniques in the setting of Kripke Open Bisimulations [14] should allow one to build a bridge between the game-semantics characterizations and Kripke Logical Relations.

Parametric bisimulations [11] have been introduced as an operational technique, merging ideas from Kripke Logical Relations and Environmental Bisimulations. They do not represent functional values coming from the environment using names, but instead use a notion of global and local knowledge to compute these values, reminiscent of the work on environmental bisimulations. The notion of global knowledge depends itself on a notion of evolving world. To our knowledge, no fully abstract Parametric Bisimulations have been presented.

A general theory of applicative [21] and normal-form bisimulations [20] has been developed, with the goal of being modular with respect to the effects considered. While the goal is similar to our work, the papers consider monadic and algebraic presentation of effects, trying particularly to design a general theory for proving soundness and completeness of such bisimulations. These works complement ours, and we would like to explore possible connections.

References

1. Abramsky, S.: Games in the semantics of programming languages. In: Proceedings of the 11th Amsterdam Colloquium. pp. 1–6. ILLC, Dept. of Philosophy, University of Amsterdam (1997)
2. Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. Information and Computation **163**, 409–470 (2000)
3. Abramsky, S., McCusker, G.: Call-by-value games. In: Proceedings of CSL. Lecture Notes in Computer Science, vol. 1414, pp. 1–17. Springer-Verlag (1997)
4. Ahmed, A., Dreyer, D., Rossberg, A.: State-dependent representation independence. In: Proceedings of POPL. pp. 340–353. ACM (2009)
5. Aristizabal, A., Biernacki, D., Lenglet, S., Polesiuk, P.: Environmental Bisimulations for Delimited-Control Operators with Dynamic Prompt Generation. Logical Methods in Computer Science **13**(3) (2017)
6. Biernacki, D., Lenglet, S.: Environmental bisimulations for delimited-control operators. In: Proceedings of APLAS. Lecture Notes in Computer Science, vol. 8301, pp. 333–348. Springer (2013)
7. Biernacki, D., Lenglet, S., Polesiuk, P.: A complete normal-form bisimilarity for state. In: Proceedings of FOSSACS. Lecture Notes in Computer Science, vol. 11425, pp. 98–114. Springer (2019)
8. Dreyer, D., Neis, G., Birkedal, L.: The impact of higher-order state and control effects on local relational reasoning. J. Funct. Program. **22**(4-5), 477–528 (2012)
9. Ghica, D.R., Tzevelekos, N.: A system-level game semantics. Electr. Notes Theor. Comput. Sci. **286**, 191–211 (2012)
10. Honsell, F., Mason, I.A., Smith, S.F., Talcott, C.L.: A variable typed logic of effects. Inf. Comput. **119**(1), 55–90 (1995)
11. Hur, C.K., Dreyer, D., Neis, G., Vafeiadis, V.: The marriage of bisimulations and kripke logical relations. In: Proceedings of POPL. pp. 59–72. ACM (2012)
12. Hyland, J.M.E., Ong, C.H.L.: On Full Abstraction for PCF: I. Models, observables and the full abstraction problem, II. Dialogue games and innocent strategies, III. A fully abstract and universal game model. Information and Computation **163(2)**, 285–408 (2000)
13. Jaber, G.: Operational nominal game semantics. In: Proceedings of FOSSACS. Lecture Notes in Computer Science, vol. 9034, pp. 264–278 (2015)
14. Jaber, G., Tabareau, N.: Kripke open bisimulation - A marriage of game semantics and operational techniques. In: Proceedings of APLAS. Lecture Notes in Computer Science, vol. 9458, pp. 271–291 (2015)
15. Jaber, G., Tzevelekos, N.: Trace semantics for polymorphic references. In: Proceedings of LICS. pp. 585–594. ACM (2016)
16. Jagadeesan, R., Pitcher, C., Riely, J.: Open bisimulation for aspects. In: Proceedings of AOSD. ACM International Conference Proceeding Series, vol. 208, pp. 107–120 (2007)
17. Jeffrey, A., Rathke, J.: Towards a theory of bisimulation for local names. In: Proceedings of LICS. pp. 56–66 (1999)
18. Jeffrey, A., Rathke, J.: A fully abstract may testing semantics for concurrent objects. Theor. Comput. Sci. **338**(1-3), 17–63 (2005)
19. Koutavas, V., Wand, M.: Small bisimulations for reasoning about higher-order imperative programs. In: Proceedings of POPL. pp. 141–152. ACM (2006)
20. Lago, U.D., Gavazzo, F.: Effectful normal form bisimulation. In: Proceedings of ESOP. Lecture Notes in Computer Science, vol. 11423, pp. 263–292. Springer (2019)

21. Lago, U.D., Gavazzo, F., Levy, P.B.: Effectful applicative bisimilarity: Monads, relators, and howe's method. In: Proceedings of LICS. IEEE Press (2017)
22. Laird, J.: Full abstraction for functional languages with control. In: Proceedings of 12th IEEE Symposium on Logic in Computer Science. pp. 58–67 (1997)
23. Laird, J.: A fully abstract trace semantics for general references. In: Proceedings of ICALP, Lecture Notes in Computer Science, vol. 4596, pp. 667–679. Springer (2007)
24. Lassen, S.B., Levy, P.B.: Typed normal form bisimulation. In: Proceedings of CSL, Lecture Notes in Computer Science, vol. 4646, pp. 283–297. Springer (2007)
25. Levy, P.B., Staton, S.: Transition systems over games. In: Proceedings of CSL-LICS. pp. 64:1–64:10 (2014)
26. Milner, R.: Fully abstract models of typed lambda-calculi. Theoretical Computer Science **4**(1), 1–22 (1977)
27. Murawski, A.S., Tzevelekos, N.: Game semantics for good general references. In: Proceedings of LICS. pp. 75–84. IEEE Computer Society Press (2011)
28. Pitts, A.M., Stark, I.D.B.: Operational reasoning for functions with local state. In: Gordon, A.D., Pitts, A.M. (eds.) Higher-Order Operational Techniques in Semantics, pp. 227–273. Cambridge University Press (1998)
29. Sangiorgi, D.: Expressing mobility in process algebras: First-order and higher-order paradigms. Tech. Rep. CST-99-93, University of Edinburgh (1993), PhD thesis
30. Sangiorgi, D., Kobayashi, N., Sumii, E.: Environmental bisimulations for higher-order languages. ACM Trans. Program. Lang. Syst. **33**(1), 5 (2011)
31. Støvring, K., Lassen, S.B.: A complete, co-inductive syntactic theory of sequential control and state. In: POPL. pp. 161–172. ACM (2007)
32. Sumii, E.: A complete characterization of observational equivalence in polymorphic *lambda*-calculus with general references. In: Proceedings of CSL. Lecture Notes in Computer Science, vol. 5771, pp. 455–469. Springer (2009)
33. Talcott, C.L.: Reasoning about functions with effects. In: Gordon, A.D., Pitts, A.M. (eds.) Higher-Order Operational Techniques in Semantics, pp. 347–390. Cambridge University Press (1998)
34. Yachi, T., Sumii, E.: A sound and complete bisimulation for contextual equivalence in λ-calculus with call/cc. In: Proceedings of APLAS. pp. 171–186. Springer (2016)

A Additional material for Section 2 (HOSC)

A.1 Type System

Please see Figure 8.

A.2 Proof of Lemma 1 (CIU)

In [10, 33], the authors propose general frameworks for establishing CIU theorems for higher-order languages with effects and control. The results are based on the usual contextual testing observing termination. Below we repeat the pattern of their argument in our framework for both $\lesssim^{\mathbf{x}(ciu)}_{ter}$ and $\lesssim^{\mathbf{x}(ciu)}_{err}$. The names of the lemmas come from Section 2.3 of [10]. Their technical aim is to establish that each relation is a precongruence.

Let $\mathbf{y} \in \{ter, err\}$.

$$\frac{}{\Sigma; \Gamma \vdash () : \mathrm{Unit}} \qquad \frac{}{\Sigma; \Gamma \vdash \mathbf{tt} : \mathrm{Bool}} \qquad \frac{}{\Sigma; \Gamma \vdash \mathbf{ff} : \mathrm{Bool}} \qquad \frac{}{\Sigma; \Gamma \vdash \widehat{n} : \mathrm{Int}}$$

$$\frac{(x, \tau) \in \Gamma}{\Sigma; \Gamma \vdash x : \tau} \qquad \frac{(\ell, \tau) \in \Sigma}{\Sigma; \Gamma \vdash \ell : \mathrm{ref}\tau} \qquad \frac{\Sigma; \Gamma \vdash M : \sigma \qquad \Sigma; \Gamma \vdash N : \tau}{\Sigma; \Gamma \vdash \langle M, N \rangle : \sigma \times \tau}$$

$$\frac{\Sigma; \Gamma \vdash M : \tau_1 \times \tau_2}{\Sigma; \Gamma \vdash \pi_i M : \tau_i} \qquad \frac{\Sigma; \Gamma, x : \sigma \vdash M : \tau}{\Sigma; \Gamma \vdash \lambda x^\sigma . M : \tau} \qquad \frac{\Sigma; \Gamma, f : \sigma \to \tau, x : \sigma \vdash M : \tau}{\Sigma; \Gamma \vdash \mathbf{rec}\ f(x^\sigma).M : \sigma \to \tau}$$

$$\frac{\Sigma; \Gamma \vdash M : \sigma \to \tau \qquad \Sigma; \Gamma \vdash N : \sigma}{\Sigma; \Gamma \vdash MN : \tau}$$

$$\frac{\Sigma; \Gamma \vdash M : \tau}{\Sigma; \Gamma \vdash \mathrm{ref}_\tau\, M : \mathrm{ref}\tau} \qquad \frac{\Sigma; \Gamma \vdash M : \mathrm{ref}\tau}{\Sigma; \Gamma \vdash !M : \tau} \qquad \frac{\Sigma; \Gamma \vdash M : \mathrm{ref}\tau \qquad \Sigma; \Gamma \vdash N : \tau}{\Sigma; \Gamma \vdash M := N : \mathrm{Unit}}$$

$$\frac{\Sigma; \Gamma \vdash M_1 : \mathrm{Bool} \qquad \Sigma; \Gamma \vdash M_2 : \tau \qquad \Sigma; \Gamma \vdash M_3 : \tau}{\Sigma; \Gamma \vdash \mathrm{if}\ M_1\ M_2\ M_3 : \tau}$$

$$\frac{\Sigma; \Gamma \vdash M_1 : \mathrm{Int} \qquad \Sigma; \Gamma \vdash M_2 : \mathrm{Int}}{\Sigma; \Gamma \vdash M_1 \oplus M_2 : \mathrm{Int}} \qquad \frac{\Sigma; \Gamma \vdash M_1 : \mathrm{Int} \qquad \Sigma; \Gamma \vdash M_2 : \mathrm{Int}}{\Sigma; \Gamma \vdash M_1 \boxdot M_2 : \mathrm{Bool}}$$

$$\frac{\Sigma; \Gamma \vdash M_1 : \mathrm{ref}\tau \qquad \Sigma; \Gamma \vdash M_2 : \mathrm{ref}\tau}{\Sigma; \Gamma \vdash M_1 = M_2 : \mathrm{Bool}}$$

$$\frac{\Sigma; \Gamma, x : \tau \vdash K[x] : \sigma}{\Sigma; \Gamma \vdash \mathrm{cont}_\tau\, K : \mathrm{cont}\ \tau} \qquad \frac{\Sigma; \Gamma, x : \mathrm{cont}\ \tau \vdash M : \tau}{\Sigma; \Gamma \vdash \mathrm{call/cc}_\tau(x.M) : \tau}$$

$$\frac{\Sigma; \Gamma \vdash M : \sigma \qquad \Sigma; \Gamma \vdash N : \mathrm{cont}\ \sigma}{\Sigma; \Gamma \vdash \mathrm{throw}_\tau\ \mathrm{M}\ \mathrm{to}\ N : \tau}$$

Fig. 8. HOSC typing rules

Lemma 14 (Op CIU). *Suppose* $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$*. Then, whenever the terms are typable and the relevant operation is allowable in an* $\mathbf{x}$*-context, we have:*

- $\langle M_1, M\rangle \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \langle M_2, M\rangle$, $\pi_i M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \pi_i M_2$, $M_1 M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 M$, $\mathrm{ref}\, M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{ref}\, M_2$, $!M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} !M_2$, $M_1 := M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 := M$, $\mathrm{if}\ M_1\ M\ M' \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{if}\ M_2\ M\ M'$, $M_1 \oplus M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 \oplus M$, $M_1 \boxdot M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 \boxdot M$, $M_1 = M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 = M$, $\mathrm{throw}\ \mathrm{M}_1\ \mathrm{to}\ M \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{throw}\ \mathrm{M}_2\ \mathrm{to}\ M$;
- $\langle M, M_1\rangle \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \langle M, M_2\rangle$, $M M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M M_2$, $M := M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M := M_2$, $\mathrm{if}\ M\ M_1\ M' \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{if}\ M\ M_2\ M'$, $\mathrm{if}\ M\ M'\ M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{if}\ M\ M'\ M_2$, $M \oplus M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M \oplus M_2$, $M \boxdot M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M \boxdot M_2$, $M = M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M = M_2$, $\mathrm{throw}\ \mathrm{M}\ \mathrm{to}\ M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{throw}\ \mathrm{M}\ \mathrm{to}\ M_2$.

Proof. We handle the first case from each category, as the rest are analogous.

- Suppose K, γ, h are such that $(K[\langle M_1, M\rangle\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.
  Observe that $K[\langle M_1, M\rangle\{\gamma\}] = K[\langle M_1\{\gamma\}, M\{\gamma\}\rangle] = K'[M_1\{\gamma\}]$ for some K'.
  Because $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$ and $(K'[M_1\{\gamma\}], h) \Downarrow_{\mathbf{y}}$, we get $(K'[M_2\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.
  Because $K[\langle M_2, M\rangle\{\gamma\}] = K[\langle M_2\{\gamma\}, M\{\gamma\}\rangle] = K'[M_2\{\gamma\}]$, this implies $(K[\langle M_2, M\rangle\{\gamma\}], h) \Downarrow_{\mathbf{y}}$, as needed.
- Suppose K, γ, h are such that $(K[\langle M, M_1\rangle\{\gamma\}]\, h) \Downarrow_{\mathbf{y}}$. We need to show $(K[\langle M, M_2\rangle\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.
  Observe that $K[\langle M, M_1\rangle\{\gamma\}] = K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle]$.
  We will argue by induction on the number of transitions in $(K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle], h) \Downarrow_{\mathbf{y}}$ for all $M\{\gamma\}, h$.
  Because of $(K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle, h) \Downarrow_{\mathbf{y}}$, we have the following cases for $M\{\gamma\}$.
  - $(M\{\gamma\} = V)$
    In this case, $K[\langle M, M_1\rangle\{\gamma\}] = K[\langle V, M_1\{\gamma\}\rangle] = K'[M_1\{\gamma\}]$.
    By $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$, we get $(K'[M_2\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.
    Because $K[\langle M_2, M\rangle\{\gamma\}] = K'[M_2\{\gamma\}]$, we obtain $(K[\langle M_2, M\rangle\{\gamma\}], h) \Downarrow_{\mathbf{y}}$, as needed.
  - $(M\{\gamma\} = K'[err()]$, only for $y = err)$
    Here $K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle]$ in 0 steps, and it follows that $(K[\langle M\{\gamma\}, M_2\{\gamma\}\rangle], h) \Downarrow_{\mathbf{y}}$.
  - $(M\{\gamma\} = K'[N]$ such that $(K'[N], h) \to (K'[N'], h'))$
    $(K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle], h) = (K[\langle K'[N]\{\gamma\}, M_1\{\gamma\}\rangle], h) \to (K[\langle K'[N']\{\gamma\}, M_1\{\gamma\}\rangle], h') \Downarrow_{\mathbf{y}}$.
    By IH, $(K[\langle K'[N']\{\gamma\}, M_2\{\gamma\}\rangle], h') \Downarrow_{\mathbf{y}}$.
    Hence, because $(K[\langle M\{\gamma\}, M_2\{\gamma\}\rangle], h) \to (K[\langle K'[N']\{\gamma\}, M_2\{\gamma\}\rangle], h')$, we have
    $(K[\langle M\{\gamma\}, M_2\{\gamma\}\rangle], h) \Downarrow_{\mathbf{y}}$.
    Note that this case also covers the reduction rule for call/cc.
  - $(M\{\gamma\} = K'[\mathrm{throw\ V\ to\ cont}\, K''])$
    In this case, $(K[\langle M\{\gamma\}, M_1\{\gamma\}\rangle], h) \to (K''[V], h)$ and $(K''[V], h) \Downarrow_{\mathbf{y}}$.
    Note that then $(K[\langle M\{\gamma\}, M_2\{\gamma\}\rangle], h) \to (K''[V], h)$ too, so we are done.

Lemma 15 (Lambda CIU). $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$ *implies* $\lambda x.M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \lambda x.M_2$.

Proof. Take K, γ, h such that $(K[(\lambda x.M_1)\{\gamma\}], h) \Downarrow_{\mathbf{y}}$. Let us write M_i^γ for $M_i\{\gamma\}$. Note that $(\lambda x.M_1)\{\gamma\} = \lambda x.M_1^\gamma$. We need to show $(K[\lambda x.M_2^\gamma], h) \Downarrow_{\mathbf{y}}$.

Instead we shall show that $M\{\lambda x.M_1^\gamma/z\} \Downarrow_{\mathbf{y}}$ implies $M\{\lambda x.M_2^\gamma/z\} \Downarrow_{\mathbf{y}}$ for any $\Sigma; z \vdash M$. The Lemma then follows by taking $M = K[z]$.

We use induction on the number of steps k in $(M\{\lambda x.M_1^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$ for all M, h.

Suppose $(M\{\lambda x.M_1^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$.

- If $k = 0$ and $y = err$ then $M = K'[err()]$. Thus, $(M\{\lambda x.M_2^\gamma/z\}, h) \Downarrow_{err}$ too.
- If $k = 0$ and $y = ter$ then $M = V\{\lambda x.M_2^\gamma/z\}$ or $M = z$. In both cases, $M\{\lambda x.M_2^\gamma/z\}$ is a value, and $M\{\lambda x.M_2^\gamma/z\} \Downarrow_{ter}$.
- Suppose $k > 0$. Because $(M\{\lambda x.M_1^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$, the following cases arise.
  - ($M = K'[N]$ and $(K'[N], h) \to (K'[N'], h')$)
    Then $(K'[N']\{\lambda x.M_1^\gamma/z\}, h') \Downarrow_{\mathbf{y}}$ in $(k-1)$ steps.
    So, by IH, $(K'[N']\{\lambda x.M_2^\gamma/z\}, h') \Downarrow_{\mathbf{y}}$.
    Because $(M\{\lambda x.M_2^\gamma/z\}, h) \to (K'[N']\{\lambda x.M_2^\gamma/z\}, h')$, we are done.
  - ($M = K'[\text{throw V to cont } K'']$)
    Then $(K''[V]\{\lambda x.M_1^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$ in $(k-1)$ steps.
    So, by IH, $(K''[V]\{\lambda x.M_2^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$.
    Because $(M\{\lambda x.M_2^\gamma/z\}, h) \to (K''[V]\{\lambda x.M_2^\gamma/z\}, h)$, we are done.
  - ($M = K'[zV]$)
    Then $(K'[M_1^\gamma\{V/x\}]\{\lambda x.M_1^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$ in $(k-1)$ steps.
    By IH, $(K'[M_1^\gamma\{V/x\}]\{\lambda x.M_2^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$.
    Because $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$, this implies $(K'[M_2^\gamma\{V/x\}]\{\lambda x.M_2^\gamma/z\}, h) \Downarrow_{\mathbf{y}}$.
    Since $(M\{\lambda x.M_2^\gamma/z\}, h) \to (K'[M_2^\gamma\{V/x\}]\{\lambda x.M_2^\gamma/z\}, h)$, we are done.

Lemma 16 (fix CIU). $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$ *implies* $\mathbf{rec}\ f(x).M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathbf{rec}\ f(x).M_2$.

Proof. Take K, γ, h such that $(K[(\mathbf{rec}\ f(x).M_1)\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.

We need to show $(K[\mathbf{rec}\ f(x).M_2)\{\gamma\}], h) \Downarrow_{\mathbf{y}}$.

Let us write M_i^γ for $M_i\{\gamma\}$, and F_i for $\mathbf{rec}\ f(x).M_i^\gamma$.

We follow the same pattern as in the previous case and show that $M\{F_1/z\} \Downarrow_{\mathbf{y}}$ implies $M\{F_2/z\} \Downarrow_{\mathbf{y}}$ for any $\Sigma; z \vdash M$. The Lemma then follows by taking $M = K[z]$.

We use induction on the number of steps k in $(M\{F_1/z\}, h) \Downarrow_{\mathbf{y}}$ for all M, h. Suppose $(M\{F_1/z\}, h) \Downarrow_{\mathbf{y}}$.

The following cases can be argued in the same way as above.

- ($M = K'[err()]$, $y = err$)
- ($M = V$ or $M = z$, $y = ter$)
- ($M = K'[N]$ and $(K'[N], h) \to (K'[N'], h')$)
- ($M = K'[\text{throw V to cont } K'']$)

It remains to deal with

- $(M = K'[zV])$
  Then $(K'[M_1^\gamma\{V/x\}\{F_1/f\}]\{F_1/z\}, h) \Downarrow_{\mathbf{y}}$ in $(k-1)$ steps.
  Observe that $(K'[M_1^\gamma\{V/x\}\{F_1/f\}]\{F_1/z\}, h) = (K'[M_1^\gamma\{V/x\}\{z/f\}]\{F_1/z\}, h)$.
  Hence, by IH, $(K'[M_1^\gamma\{V/x\}\{z/f\}]\{F_2/z\}, h) \Downarrow_{\mathbf{y}}$.
  Because $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$, this implies $(K'[M_2^\gamma\{V/x\}\{z/f\}]\{F_2/z\}, h) \Downarrow_{\mathbf{y}}$.
  Since $(M\{F_2/z\}, h) \to (K'[M_2^\gamma\{V/x\}\{z/f\}]\{F_2/z\}, h)$, we are done.

Lemma 17 (call/cc CIU). $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$ *implies* $\mathrm{call/cc}(x.M_1) \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} \mathrm{call/cc}(x.M_2)$.

Proof. Let K, γ, h be such that $(K[\mathrm{call/cc}(x.M_1)], h) \Downarrow_{\mathbf{y}}$. Note that

$$(K[\mathrm{call/cc}(x.M_1)\{\gamma\}], h) \to (K[M_1\{\mathrm{cont}\, K/x\}]\{\gamma\}], h) = (K[M_1\{\gamma\cdot[x \mapsto \mathrm{cont}\, K]\}], h).$$

Because of $M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2$, we get $(K[M_2\{\gamma\cdot[x \mapsto \mathrm{cont}\, K]\}], h) \Downarrow_{\mathbf{y}}$ Consequently, $(K[\mathrm{call/cc}(x.M_2)\{\gamma\}], h) \Downarrow_{\mathbf{y}}$, because $(K[\mathrm{call/cc}(x.M_2)\{\gamma\}], h) \to (K[M_2\{\gamma \cdot [x \mapsto \mathrm{cont}\, K]\}], h)$.

Lemma 18 (Precongruence). *Suppose* $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$*,* $\Gamma \vdash M_1, M_2 : \sigma$ *are* HOSC*-terms with an* $\mathbf{x}$ *boundary, and* C *is an* $\mathbf{x}$*-context such that* $\Gamma' \vdash C[M_1], C[M_2] : \sigma'$*. Then* $\Gamma \vdash M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 : \sigma$ *implies* $\Gamma' \vdash C[M_1] \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} C[M_2] : \sigma'$*.*

Proof. By induction on the structure of contexts using the preceding lemmas.

Corollary 4 (CIU result). *Suppose* $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}, \mathrm{HOS}, \mathrm{GOS}\}$ *and* $\Gamma \vdash M_1, M_2 : \sigma$ *are* HOSC*-terms with an* $\mathbf{x}$ *boundary.* $\Gamma \vdash M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}(ciu)} M_2 : \sigma$ *iff* $\Gamma \vdash M_1 \lesssim_{\mathbf{y}}^{\mathbf{x}} M_2 : \sigma$*.*

Proof. The left-to-right implication follows from Lemma 18. The right-to-left implication holds, because testing with h, K, γ is a special case of testing with C.

The Corollary is the same as Lemma 1.

B Additional material for Section 3 (HOSC[HOSC])

B.1 Extended Operational Semantics

Definition 20. *Taking* M *a term,* c *a continuation name,* h *a heap we write* $\Sigma; \Gamma \vdash (M, c, h) : \tau$ *if* $\Sigma; \Gamma \vdash M : \tau$*,* $c : \tau$ *and* $h : (\Sigma; \Gamma)$*.*

Lemma 19. *Taking* $\Sigma; \Gamma \vdash (M, c, h) : \tau$*, then:*

- *either* (M, c, h) *is reducible (for* $\to$*);*
- *or* M *is a a callback* $K[f\ V]$ *with* $f \in \mathrm{dom}(\Gamma)$*;*
- *or* M *is a value* V*.*

Lemma 20. *Taking $\Sigma;\Gamma \vdash (M,c,h):\tau$, and $\Sigma;\Gamma \vdash K \div \tau$ and γ an idempotent substitution s.t. $\vdash \gamma : \Gamma$, writing $\tilde{M}$ for $M\{\gamma\}$ and $\tilde{h}$ for $h\{\gamma\}$ then $(K[\tilde{M}],\tilde{h}) \to (N,h')$ implies that*

- *either $(M,c,h) \to (M',c',h'')$ and $N = K'[M'\{\gamma\}]$ with $K' = \gamma(c')$, and $h''\{\gamma\} = h'$;*
- *or M is a callback $K'[f\ V]$ with $\gamma(f)$ a λ-abstraction $\lambda x.P$ and $N = K[\tilde{K'}[P\{\tilde{V}/x\}]]$, with $\tilde{K'} = K'\{\gamma\}$ and $\tilde{V} = V\{\gamma\}$;*
- *or M is a value and K is an evaluation context larger than $\bullet$.*

Definition 21. *Taking M an extended term and κ a substitution from continuation names to evaluation contexts that contains the continuation names appearing in the support of M, one write $M\{\kappa\}$ for the term where all the occurrences of $\mathrm{cont}\,K,c$ are substituted by $\mathrm{cont}\,K'[K[\bullet]]$, with $\kappa(c) = K'$. One extend this definition to heaps, writing $h\{\kappa\}$ for the heap $\{(\ell, v\{\kappa\}) \mid (\ell,v) \in h\}$.*

Theorem 9. *Taking M a term, h a heap, κ a substitution from continuation names to evaluation contexts that contains the continuation names appearing in the support of M and h, and c,c' two continuation names s.t. $\kappa(c) = K$ and $\kappa(c') = K'$, then for all M', h', if $(M,c,h) \to (M',c',h')$ then $(K[M\{\kappa\}], h\{\kappa\}) \to (K'[M'\{\kappa\}], h'\{\kappa\})$.*

Proof. We reason by case analysis:

- if $M = K_1[\mathrm{call/cc}(x.M_1)]$, then one has:
  - $(K_1[\mathrm{call/cc}(x.M_1)], c, h) \to (K_1[M_1\{\mathrm{cont}\,K_1, c/x\}], c, h)$;
  - $(K[K_1[\mathrm{call/cc}(x.M_1)]\{\kappa\}], h\{\kappa\}) \to (K[K_1[M_1\{\mathrm{cont}\,K[K_1]/x\}]\{\kappa\}], h\{\kappa\})$

  and we conclude using the fact that $(M_1\{\mathrm{cont}\,K_1, c/x\})\{\kappa\} = (M_1\{\mathrm{cont}\,K[K_1]/x\})\{\kappa\}$ since $\kappa(c) = K$.
- if $M = K_1[\text{throw V to cont}\,K_2, c']$, then one has:
  - $(K_1[\text{throw V to cont}\,K_2, c'], c, h) \to (K_2[V], c', h)$;
  - $(K[K_1[\text{throw V to cont}\,K_2, c']\{\kappa\}], h\{\kappa\}) \to (K'[K_2[V]\{\kappa\}], h\{\kappa\})$ since $\kappa(c') = K'$.
- If there exists a (unique) reduction $(M,h) \to (M',h')$ then:
  - $(M\{\kappa\}, c, h\{\kappa\}) \to (M'\{\kappa\}, c, h'\{\kappa\})$
  - $(K[M\{\kappa\}], h\{\kappa\}) \to (K[M'\{\kappa\}], h'\{\kappa\})$.

B.2 Proof of Lemma 2

Proof. We reason by contraposition.

1. Suppose $\Gamma \vdash M_1 \not\lesssim^{\mathbf{x}}_{ter} M_2 : \tau$, i.e. $C[M_1] \Downarrow_{ter}$ and $C[M_2] \not\Downarrow_{ter}$ for some $\vdash C \div \tau$.

   Then we can construct $err \vdash C' \div \tau$ such that $C'[M_1] \Downarrow_{err}$ and $C'[M_2] \not\Downarrow_{err}$ as follows:
   $$C'[\bullet] = (C_{;err}[\bullet]; err),$$
   where $C_{;err}$ refers to C in which each occurrence of $\mathrm{cont}_\sigma\,(-)$ is replaced with $\mathrm{cont}_\sigma\,(-; err)$. In this way, the construction transforms all opportunities for

$\Downarrow_{ter}$ into ones for $\Downarrow_{err}$. Note that, if M_1 contained $\mathrm{cont}_\sigma\, K$, it would not necessarily be the case that $C'[M_1] \Downarrow_{err}$, because M_1 is not affected by the transformation.

2. Let $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}\}$. Suppose $\Gamma \vdash M_1 \not\lesssim^{\mathbf{x}}_{err} M_2$, i.e. $C[M_1] \Downarrow_{err}$ and $C[M_2] \not\Downarrow_{err}$ for some C such that $err \vdash C \div \tau$. Then we can construct $\vdash C' \div \tau$ such that $C'[M_1] \Downarrow_{ter}$ and $C'[M_2] \not\Downarrow_{ter}$ as follows.

$$C'[\bullet] = \mathrm{call/cc}(y.\, C_{;\Omega}[\bullet]\{(\lambda z.\mathrm{throw}\ ()\ \mathrm{to}\ y)/err\}; \Omega)$$

where $C_{;\Omega}$ is defined analogously to $C_{;err}$. Note that we add $;\Omega$, because $C[M_2] \not\Downarrow_{err}$ could be due to $\Downarrow_{ter}$ (rather than divergence), and we want to make sure that $C'[M_2]$ diverges, which will imply $C'[M_2] \not\Downarrow_{ter}$.
Note that, because of the use of continuations, C' is an $\mathbf{x}$-context only for $\mathbf{x} \in \{\mathrm{HOSC}, \mathrm{GOSC}\}$.
In this case, we also rely on $\mathrm{cont}_\sigma\,(K)$-freeness (of M_2). If $C[M_2] \not\Downarrow_{err}$ was due to $\Downarrow_{ter}$ caused by $\mathrm{cont}_\sigma\, K$ in M_2, then our $;\Omega$ transformation might not imply divergence for $C'[M_2]$.

B.3 Name invariance

We say that a permutation p of Names is ***type-preserving*** if it is also a permutation once restricted to each of CNames_σ and $\mathrm{FNames}_{\sigma\to\sigma'}$. Given $X \subseteq \mathrm{Names}$, we say that p fixes X if $p(x) = x$ for all $x \in X$. Type-preserving permutations can be applied to traces in the obvious way. In particular, if t is (N_O, N_P)-trace then $p(t)$ is a $(p(N_O), p(N_P))$-trace. We write $t_1 \sim_X t_2$ if there exists a type-preserving permutation p that fixes X such that $p(t_1) = t_2$.

Lemma 21. *Suppose $\mathbf{C} = \langle \cdots, \phi, h\rangle$ is a configuration and p is a type-preserving permutation. If $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ and p fixes ϕ then $p(t) \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$.*

Due to the arbitrariness of name choice in transitions (i.e. freedom to choose fresh names), $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ is closed under renamings that preserve types and the names already present in $\mathbf{C}$.

B.4 Proof of Lemma 4

Delegated to Section C.

B.5 Proof of Theorem 1

Proof. Suppose $\mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_2)$. We handle $\Gamma \vdash M_1 \lesssim^{\mathrm{HOSC}(ciu)}_{err} M_2$, as it is slightly more involved. The reasoning for $\lesssim^{\mathrm{HOSC}(ciu)}_{ter}$ is symmetric.

Let Σ, h, K, γ be such that $(K[M_1\{\gamma\}], h) \Downarrow_{err}$. Suppose $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$ and $c : \sigma'$ ($c \notin \circ$). By Lemma 4 (left-to-right), there exist t, c' such that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}}, c}_{M_1})$ and $t^\perp \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma})$. By $\mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOSC}}(\Gamma \vdash M_2)$, we have $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}}, c}_{M_2})$. Because $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}}, c}_{M_2})$ and $t^\perp \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i}, c}_{h,K,\gamma})$, by Lemma 4 (right-to-left) we can conclude $(K[M_2\{\gamma\}], h) \Downarrow_{err}$. Thus, $\Gamma \vdash M_1 \lesssim^{\mathrm{HOSC}(ciu)}_{err} M_2$.

B.6 Proof of Lemma 5

Recall that abstract values are tuples consisting of boolean and integer constants, as well as function names. We can refer to them using projections of the form $\pi_{\vec{i}}$, where $\vec{i} \in \{1,2\}^+$, on the understanding that $\pi_{i,\vec{i}}x = \pi_i(\pi_{\vec{i}}x)$.

– Suppose $\mathsf{Num}(A) = \{(\vec{i}, n) \mid \pi_{\vec{i}}A = n : \mathrm{Bool}, \mathrm{Int}\}$. Then $assert(x \sim A)$ will act as shorthand for the following code if $(\bigwedge_{(\vec{i},n)\in\mathsf{Num}(A)} \pi_{\vec{i}}x = n)$ () Ω. which checks if the boolean/integer arguments match those of A.
– Another operation, written $A[\pi x/f]$, will substitute for each $f \in \nu(A)$, the corresponding projection $\pi_{\vec{i}_f}x$ (i.e. one such that $\pi_{\vec{i}_f}A = f$).

This syntax will be used in all definability arguments.

Lemma 5 follows from the lemma given below for $i = 0$. Consider $h' = h_0$, $K' = \gamma_0(c)$, $\gamma' = \gamma_0 \setminus c$. We have $\nu(\mathrm{img}(\gamma_0), \mathrm{img}(h_0)) \subseteq \circ \uplus \{\diamond\}$. As names $\circ_\sigma$ can only occur inside terms of the form $\mathrm{cont}\,(K', \circ_\sigma)$, we can conclude that $(h', K', \gamma') = (h_\circ, K_\circ, \gamma_\circ)$, where h, K, γ are from HOSC.

Lemma 22. *Suppose* $\phi \uplus \{\diamond\} \subseteq$ FNames, $c \in$ CNames *and* $t = o_1p_1 \cdots o_np_n$ *is a* $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. Given* $0 \le i \le n$*, let* $t_i = o_{i+1}p_{i+1} \cdots o_np_n$*. There exist passive configurations* $\mathbf{C}_i$ *such that* $\mathbf{Tr}^{\mathrm{even}}(\mathbf{C}_i)$ *consists of even-length prefixes of* $o_{i+1}p_{i+1} \cdots o_np_n$ *(along with their renamings via permutations on* Names *that fix* ϕ_i*). Moreover,* $\mathbf{C}_i = \langle \gamma_i, \xi_i, \phi_i, h_i \rangle$ *(*$0 \le i \le n$*), where*

– $\mathrm{dom}(\gamma_i)$ *consists of* $\phi \cup \{c\}$ *and all names introduced by* P *in* $o_1p_1 \cdots o_ip_i$*;*
– $\nu(\mathrm{img}(\gamma_i)) = \emptyset$*;*
– $\mathrm{dom}(\xi_i)$ *consists of* c *and all continuation names introduced by* P *in* $o_1p_1 \cdots o_ip_i$*;*
– *for all* $d \in \mathrm{dom}(\xi_i)$*,* $\xi_i(d) = top_P(o_1 \cdots o_j)$ *if* d *was introduced in* p_j *(we regard* c *as being introduced in* p_0 *and define* $top_P(o_1 \cdots o_0) = \circ_{\tau'}$*);*
– ϕ_i *consists of* $\circ \uplus \{\diamond\} \uplus \phi \uplus \{c\}$ *and all names introduced in* $o_1p_1 \cdots o_ip_i$*;*
– $\mathrm{dom}(h_i) = \mathrm{dom}(h_0)$*;*
– $\nu(\mathrm{img}(h_i))$ *may only contain elements of* $\circ \uplus \{\diamond\}$ *and names introduced by* O *in* $o_1p_1 \cdots o_ip_i$*.*

Proof. The main idea is to use references in order to record all continuations and functions introduced by O, so that they can be accessed in terms at the time when they need to be used by P. Other references will also be used to inject the right pieces of code into the LTS.

Below we explain how the content of $\mathbf{C}_i$ is meant to evolve and what invariants will be maintained by the construction for each kind of name in t.

FNames **from P** Suppose n_{FP} is the number of function names in ϕ and those introduced by P in t. We shall write f_P^j $(0 \le j < n_{FP})$ to refer to the jth such name, on the understanding names from ϕ are introduced first and this is followed by names in t in order of appearance (from left to right).

For each $f_P^j : \sigma_j \to \tau_j$, we will have a dedicated reference $\mathit{fpr}_j : \mathrm{ref}(\sigma_j \to \tau_j)$ in all heaps. The content of $h_i(\mathit{fpr}_j)$ will be changing at each step of the construction and it will be used to arrange for suitable behaviour following O-actions of the form $f_P^j(A, c)$. For example, if the action is not meant to generate a response at a stage, we can use $\mathit{fpr}_j := \lambda x.(!\mathit{fpr}_j)x$ to cause divergence by creating a cycle in the heap.

If f_P^j was introduced in p_i (we take $i = 0$ for $f_P^j \in \phi$), then f_P^j will be present in all $\phi_{i'}, \gamma_{i'}$ for $i' \geq i$. We shall maintain the invariant $\gamma_{i'}(f_P^j) = \lambda x.(!\mathit{fpr}_j)x$ for all $i' \geq i$.

Note that this is consistent with $\nu(\mathrm{img}(\gamma_i)) = \emptyset$.

CNames **from P** Suppose n_{CP} is the number of continuation names introduced by P in t plus 1, to take c into account. Similarly to the previous case, we write c_P^j $(0 \leq j < c_{FP})$ to refer to the jth such name, on the understanding that $c_P^0 = c$ and other names are enumerated in the same order as they appear in t (from left to right).

For each $c_P^j : \sigma_j$, we will have a dedicated reference $\mathit{cpr}_j : \mathrm{ref}(\sigma_j \to \tau_j)$, if c_P^j was introduced in $p_{j'}$ and $\mathit{top}_O(o_1 \cdots o_{j'}) : \tau_j$, in all heaps.

Its content will be changing at each step of the construction, in order to provide suitable reactions to O-actions of the form $c_P^j(A)$.

If c_P^j was introduced in p_i (we take $i = 0$ for $c_P^j = c_P$) then c_P^j will be present in all $\phi_{i'}, \gamma_{i'}$ for $i' \geq i$. We shall maintain the invariant $\gamma_{i'}(c_P^j) = (\lambda x.(!\mathit{cpr}_j)x)\bullet$ and $\xi_{i'}(c_P^j) = \mathit{top}_O(o_1 \cdots o_{j'})$, if c_P^j was introduced in $p_{j'}$.

Note that this is consistent with $\nu(\mathrm{img}(\gamma_i)) = \emptyset$.

FNames **from O** We use similar notation here and suppose n_{FO} is the number of function names introduced by O. As in previous cases, we use f_O^j $(0 \leq j < n_{FO})$ to refer to such names.

For each $f_O^j : \sigma_j \to \tau_j$, we will have a corresponding reference $\mathit{for}_j : \mathrm{ref}(\sigma_j \to \tau_j)$ in all heaps, which will be used to store the name as soon as it is played, i.e. if f_O^j is introduced in o_i (for $\diamond$ we take $i = 0$), then $h_{i'}(\mathit{for}_j) = f_O^j$ for all $i' \geq i$. Earlier we will use a divergent value, i.e. $h_{i'}(\mathit{for}_j) = \lambda x.(!\mathit{for}_j)x$ for $i' < i$.

f_O^j will be part of $\phi_{i'}$ for all $i' \geq i$.

Note that this is consistent with: $\nu(\mathrm{img}(h_i))$ may only contain elements of $\circ \uplus \{\diamond\}$ and names introduced by O in $o_1 p_1 \cdots o_i p_i$.

CNames **from O** Suppose n_{CO} is the number of continuation names introduced by O in t. As before, we use c_O^j $(0 \leq j < n_{FO})$ to refer to such names.

For each $c_O^j : \sigma_j$, we will have a corresponding reference $\mathit{cor}_j : \mathrm{ref}(\mathrm{cont}\ \sigma_j)$, which will be used to store the name as soon as it is played, i.e. if c_O^j is introduced in o_i, then $h_{i'}(\mathit{cor}_j) = \mathrm{cont}\,(\bullet, c_O^j)$ for all $i' \geq i$. Earlier we will use a divergent value, i.e. $h_{i'}(\mathit{cor}_j) = \mathrm{cont}\,((\lambda x.\Omega)\bullet, \circ_{\tau'})$ for $i' < i$, where Ω is a divergent term.

c_O^j will be part of $\phi_{i'}$ for all $i' \geq i$.

Note that this is consistent with: $\nu(\mathrm{img}(h_i))$ may only contain elements of $\circ \uplus \{\diamond\}$ and names introduced by O in $o_1 p_1 \cdots o_i p_i$.

Overall, for each $0 \leq i \leq n$, we shall have

$$\mathrm{dom}(h_i) = \{\mathit{fpr}_j \mid 0 \leq j < n_{FP}\} \cup \{\mathit{cpr}_j \mid 0 \leq j < n_{CP}\} \cup \{\mathit{for}_j \mid 0 \leq j < n_{FO}\} \cup \{\mathit{cor}_j \mid 0 \leq j < n_{CO}\}.$$

The above description specifies ϕ_i, γ_i, ξ_i,dom(h_i) and $h_i(\mathit{for}_j)$ $(0 \leq j < n_{FO})$, $h_i(\mathit{cor}_j)$ $(0 \leq j < n_{CO})$, for any $0 \leq i \leq n$. Hence, in the forthcoming argument we will focus on defining $h_i(\mathit{fpr}_j)$ $(0 \leq j < n_{FP})$ and $h_i(\mathit{cpr}_j)$ $(0 \leq i < n_{CP})$. Because the values written to these references will only contain elements from $\circ \uplus \{\diamond\}$, it will follow that $\nu(\mathrm{img}(h_i))$ may only contain elements of $\circ \uplus \{\diamond\}$ and names introduced by O in $o_1 p_1 \cdots o_i p_i$.

We proceed by reverse induction, starting from $i = n$.

$\mathbf{i = n}$ To complete the definition of $\mathbf{C}_n$, it suffices to specify $h_n(\mathit{fpr}_j)$ $(0 \leq j < n_{FP})$ and $h_n(\mathit{cpr}_j)$ $(0 \leq j < n_{CP})$. We set $h_n(\mathit{fpr}_j) = (\lambda x.(!\mathit{fpr}_j)x)$ and $h_n(\mathit{cpr}_j) = (\lambda x.(!\mathit{cpr}_j)x)$, i.e. deferencing will cause divergence. Consequently, because $\gamma_n(f_P^j) = \lambda x.(!\mathit{fpr}_j)x$ and $\gamma_n(c_P^j) = \lambda x.(!\mathit{cpr}_j)x$, any O action from $\mathbf{C}_n$ will trigger divergence. Thus, the only even-length trace that can be generated is the empty one, and we have $\mathbf{Tr}^{\mathrm{even}}(\mathbf{C}_n) = \{\epsilon\}$, as required.

$\mathbf{0 \leq i < n}$ Let $0 \leq i < n$. Assume validity of the Lemma for $i+1$ and suppose $\mathbf{C}_{i+1} = \langle \gamma_{i+1}, \xi_{i+1}, \phi_{i+1}, h_{i+1} \rangle$. By case analysis on p_{i+1}, we first construct an active configuration $E_i = \langle M', c', \gamma_i', \xi_i', \phi_i', h_{i+1} \rangle$ such that $E_i \xrightarrow{p_{i+1}} \mathbf{C}_{i+1}$.

Given an abstract value A, let $V_A = A[(\lambda x.(!\mathit{fpr}_j)x)/f_P^j]$, i.e. the function names f_P^j are replaced with function values $(\lambda x.(!\mathit{fpr}_j)x)$. Below we write $\phi_{i+1} \setminus X$, $\gamma_{i+1} \setminus X$ and $\xi_{i+1} \setminus X$ to stand for the removal of names in X from the domain of the respective function, while preserving values for other elements. The table below shows the components of E_i in each case.

p_{i+1}	M'	c'	γ_i'	ξ_i'	ϕ_i'
$\bar{c}_O^{j'}(A)$	V_A	$c_O^{j'}$	$\gamma_{i+1} \setminus A$	ξ_{i+1}	$\phi_{i+1} \setminus A$
$\bar{f}_O^{j'}(A, c_P^{j''})$	$(\lambda x.(!\mathit{cpr}_{j''})x)[f_O^{j'} V_A]$	$\mathit{top}_P(o_1 \cdots o_{i+1})$	$\gamma_{i+1} \setminus A, c_P^{j''}$	$\xi_{i+1} \setminus c_P^{j''}$	$\phi_{i+1} \setminus A, c_P^{j''}$

Note that, in each case, $E_i \xrightarrow{p_{i+1}} \mathbf{C}_{i+1}$. In particular, our definition of V_A (based on $\lambda x.(!\mathit{fpr}_j)x$) and the occurrence of $\lambda x.(!\mathit{cpr}_{j'})x$ in the second case guarantee that, after the step, γ_i' extends to γ_{i+1} in accordance with our description of γ_{i+1} at the beginning of the proof. Similarly, setting c' to $\mathit{top}_P(o_1 \cdots o_{i+1})$ in the second case means that ξ_i' will evolve into ξ_{i+1}.

As a next step we define another active configuration $D_i = \langle M'', \mathit{top}_P(o_1 \cdots o_{i+1}), \gamma_i', \xi_i', \phi_i', h_{i+1} \rangle$, where M'' is specified by the table below, by case analysis on p_{i+1}.

Note that $D_i \xrightarrow{\tau} E_i$.

p_{i+1}	M''	
$c_O^{j'}(A)$	throw $\mathrm{V_A}$ to cont $(\bullet, \circ_\sigma)$	$c_O^{j'} = \circ_\sigma$
$c_O^{j'}(A)$	throw $\mathrm{V_A}$ to $!cor_{j'}$	$c_O^{j'} \notin \circ$
$f_O^{j'}(A, c_P^{j''})$	$(\lambda x.(!cpr_{j''})x)(\diamond V_A)$	$f_O^{j'} = \diamond$
$f_O^{j'}(A, c_P^{j''})$	$(\lambda x.(!cpr_{j''})x)((!for_{j'})V_A)$	$f_O^{j'} \neq \diamond$

Finally, we are ready to define $\mathbf{C}_i = \langle \gamma_i, \xi_i, \phi_i, h_i \rangle$ by case analysis on o_{i+1}. Recall that ϕ_i, γ_i, ξ_i, $\mathrm{dom}(h_i)$, $h_i(\mathit{for}_j)$ $(0 \leq j < n_{FO})$, $h_i(\mathit{cor}_j)$ $(0 \leq j < n_{CO})$ are covered by the invariants discussed at the beginning of the proof. Thus, it suffices to specify $h_i(\mathit{fpr}_j)$ and $h_i(\mathit{cpr}_j)$.

- Suppose $o_{i+1} = c_P^j(A)$. Since o_{i+1} is the only O-move that should be responded to by P:
  - we let $h_i(\mathit{fpr}_{j'}) = \lambda x.(!\mathit{fpr}_{j'})x$ for any $0 \leq j' < n_{FP}$, in order to create divergence after any $f_P^{j'}(A, c_O^{j''})$;
  - we let $h_i(\mathit{cpr}_{j'}) = \lambda x.(!\mathit{cpr}_{j'})x$ for any $0 \leq j' < n_{CP}$ such that $j' \neq j$, in order to create divergence after $c_P^{j'}(A)$ with $j' \neq j$.

  To allow a suitable response after $c_P^j(A)$, we set

  $$h_i(\mathit{cpr}_j) = \lambda x.\mathit{assert}(x \sim A);\ \mathit{savefun}(A);\ \mathit{setheap}(i+1);\ M''$$

  where the special code fragments are explained below.
  - $\mathit{savefun}(A)$ is meant to save all functions from A in the corresponding references. Let $\mathsf{Fun}(A) = \{(\vec{i}, w) \mid \pi_{\vec{i}} A = f_O^w\}$. Then $\mathit{savefun}(A)$ is the sequence of assignments $\mathit{for}_w := \pi_{\vec{i}} x$, for all $(\vec{i}, w) \in \mathsf{Fun}(A)$.
  - $\mathit{setheap}(i+1)$ is the sequence of assignments $\mathit{fpr}_{j'} := h_{i+1}(\mathit{fpr}_{j'})$ $(0 \leq j' < n_{FP})$ and $\mathit{cpr}_{j'} := h_{i+1}(\mathit{cpr}_h)$ $(0 \leq j' < n_{CP})$.

  Suppose c_P^j was introduced in $p_{j'}$ then we have $\mathit{top}_P(o_1 \cdots o_{j'}) = \mathit{top}_P(o_1 \cdots o_{i+1})$, i.e. types of the codomains of $!\mathit{cpr}_j$ and $!\mathit{cpr}_{j''}$ match, and indeed we can use M'' to define $h_i(\mathit{cpr}_j)$ (note that throw is not causing typing problems). Then we have $\mathbf{C}_i \xrightarrow{o_{i+1}} C_i$, where $C_i = \langle (\lambda x.!\mathit{cpr}_j x)[A], \mathit{top}_P(o_1 \cdots o_{j'}), \gamma_i', \xi_i', \phi_i', h_i) \rangle$ and $C_i \xrightarrow{\tau^*} D_i = \langle M'', \mathit{top}_P(o_1 \cdots o_{i+1}), \gamma_i', \xi_i', , \phi_i', h_{i+1} \rangle$. Recall that we have already established $D_i \xrightarrow{\tau} E_i \xrightarrow{p_i} \mathbf{C}_{i+1}$, so we are done.
- Suppose $o_{i+1} = f_P^j(A, c_O^{j'})$. Then we let $h_i(\mathit{cpr}_{j''}) = \lambda x.(!\mathit{cpr}_{j''})x$ $(0 \leq j'' < n_{CP})$ to create divergence after any $c_P^{j''}(A)$, and $h_i(\mathit{fpr}_{j''}) = \lambda x.(!\mathit{fpr}_{j''})x$ for any $0 \leq j'' < n_{FP}$ such that $j'' \neq j$, to create divergence after any $f_P^{j''}(A, c_O^{j'''})$ with $j'' \neq j$. Then, to arrange for the right reaction after o_{i+1}, we set

  $$h_i(\mathit{fpr}_j) = \lambda x.\mathit{assert}(x \sim A);\ \mathit{savefun}(A);\ \mathrm{call/cc}(y.\mathit{cr}_{j'} := y;\ \mathit{setheap}(i{+}1);\ M'')$$

  where the special code fragments are specified above. Note that, similarly, we have $\mathbf{C}_i \xrightarrow{o_{i+1}} C_i$, where $C_i = \langle (\lambda x.!\mathit{fpr}_j x)[A], c_O^{j'}, \gamma_i, \xi_i, \phi_i, h_i) \rangle$, $C_i \xrightarrow{\tau^*} D_i =$

$\langle M'', top_P(o_1 \cdots o_{i+1}), \gamma_i', \xi_i', \phi_i', h_{i+1}\rangle$ and $D_i \xrightarrow{\tau} E_i \xrightarrow{p_{i+1}} \mathbf{C}_{i+1}$, because in this case $c_O^{j'} = top_P(o_1 \cdots o_{i+1})$.

The invariance property follows from Remark 2.

B.7 Proof of Theorem 2

Proof. Suppose $\Gamma \vdash M_1 \lesssim^{\mathrm{HOSC}}_{ciu,err} M_2$. Let ρ be a Γ-configuration, $A_i = \rho(x_i)$, $c : \sigma$ and $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_1})$. Then t is a $(\nu(\rho) \uplus \{c\}, \emptyset)$-trace. Let $t_1 = t\{\diamond'/\diamond, \circ'/\circ\}$, where $\diamond', \circ'$ are fresh names of the same type as $\diamond, \circ$ respectively (this is done to ensure that $\diamond, \circ$ do not occur in t_1). By Lemma 21, because $t_1 \sim_{\nu(\rho)\uplus\{c\}} t$, we also have $t_1 \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_1})$. Let c' : Unit be fresh. Then $t_2 = t_1^{\perp}\,\bar{\diamond}((), c')$ is an $(\{\diamond, \circ\}, \nu(\rho) \uplus \{c\})$-trace. By Lemma 5, there exists a passive configuration $\mathbf{C}_O = \langle \gamma_O, \{c \mapsto \circ\}, \nu(\rho) \uplus \{c, \diamond, \circ\}, h\rangle$ such that $\mathbf{Tr}^{\mathrm{even}}_{\mathrm{HOSC}}(\mathbf{C}_O)$ consists of all (even-length prefixes of) traces t' such that $t' \sim_{\nu(\rho)\uplus\{c,\diamond,\circ\}} t_2$. Observe that $\mathbf{C}_O = \mathsf{C}^{\vec{\gamma_i},c}_{h,K,\gamma}$, where $K = (\gamma_O(c))\{err/\diamond\}$, $\gamma(x_i) = (A_i\{\gamma_O\})\{err/\diamond\}$, and $\gamma_i = \gamma_O \upharpoonright \nu(A_i)$. Hence, $t_1 \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_1})$ and $t_1^{\perp}\,\bar{\diamond}((), c') \in \mathsf{C}^{\vec{\gamma_i},c}_{h,K,\gamma}$. By Lemma 4 (right-to-left), $(K[M_1\{\gamma\}], h) \Downarrow_{err}$. Because $\Gamma \vdash M_1 \lesssim^{\mathrm{HOSC}}_{ciu,err} M_2$, $(K[M_2\{\gamma\}], h) \Downarrow_{err}$ follows. By Lemma 4 (left-to-right), there exist t'', c'' such that $t'' \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$ and $(t'')^{\perp}\,\bar{\diamond}((), c'') \in \mathsf{C}^{\vec{\gamma_i},c}_{h,K,\gamma}$. By the definition of $\mathbf{C}_O$, we must have $(t'')^{\perp}\,\bar{\diamond}((), c'') \sim_{\nu(\rho)\uplus\{c,\diamond,\circ\}} t_1^{\perp}\,\bar{\diamond}((), c')$, so $t'' \sim_{\nu(\rho)\uplus\{c,\diamond,\circ\}} t_1$. Because $t'' \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$, we have $t_1 \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$ by Lemma 21. Since $t_1 \sim_{\nu(\rho)\uplus\{c\}} t$, it follows that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$, as required.

C Composite Interaction (Proof of Lemma 4)

Definition 22. *A* composite configuration $\mathbf{D}$ *is a tuple* $\langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *with* M *a term,* c *a continuation name,* γ_P, γ_O *two environments,* ϕ *a set of names and* h_P, h_O *two heaps.*

Definition 23. *Taking a continuation function* ξ*, we define a relation* $\prec_\xi$ *between the continuation names as the graph of* ξ*, i.e.* $c \prec_\xi c'$ *when* $\xi(c) = c'$*.*

We write $\circ$ for the final continuation name, used by Opponent to answer the resulting value of the whole interaction.

Definition 24. *A* valid composite configuration $\mathbf{D}$ *is a tuple* $\langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *with:*

- $\mathrm{dom}(\gamma_P) \cap \mathrm{dom}(\gamma_O) = \varnothing$ *and* $\circ \notin \mathrm{dom}(\gamma_P) \cup \mathrm{dom}(\gamma_O)$*;*
- $\mathrm{dom}(\gamma_P) \cup \mathrm{dom}(\gamma_O) \cup \{\circ, \diamond\} = \phi$*;*
- $\mathrm{dom}(\xi) = (\mathrm{dom}(\gamma_O) \cup \mathrm{dom}(\gamma(P))) \cap \mathrm{CNames}$*;*
- *for all* $c \in \mathrm{dom}(\xi)$*, if* $c \in \mathrm{dom}(\gamma_X)$ *then* $\xi(c) \in \mathrm{dom}(\gamma_{X^\perp})$*, for* $X \in \{O, P\}$*;*
- *the transitive closure of* $\prec_\xi$ *is a strict partial order which admit a unique maximal element equal to* $\circ$*;*

$$
\begin{array}{l|lcl}
(P\tau) & \langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{\tau} & \langle N, c', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O \rangle \\
& \multicolumn{3}{l}{\text{when } c \in \mathrm{dom}(\gamma_O) \text{ and } (M, c, h_P) \to (N, c', h'_P)} \\
(PA) & \langle V, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{\bar{c}(A)} & \langle K[A], \xi(c), \gamma_P \cdot \gamma', \gamma_O, \xi, \phi \uplus \mathrm{dom}(\gamma'), h_P, h_O \rangle \\
& \multicolumn{3}{l}{\text{when } c : \sigma, \gamma_O(c) = K, \text{ and } (A, \gamma') \in \mathbf{AVal}_\sigma(V)} \\
(PQ) & \langle K[fV], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{\bar{f}(A,c')} & \langle V'A, c', \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \gamma_O, \xi \cdot [c' \mapsto c], \\
& & & \quad \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h_P, h_O \rangle \\
& \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma', c' : \sigma', \gamma_O(f) = V' \text{ and } (A, \gamma') \in \mathbf{AVal}_\sigma(V)} \\
(O\tau) & \langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{\tau} & \langle N, c', \gamma_P, \gamma_O, \xi, \phi, h_P, h'_O \rangle \\
& \multicolumn{3}{l}{\text{when } c \in \mathrm{dom}(\gamma_P) \text{ and } (M, c, h_O) \to (N, c', h'_O)} \\
(OA) & \langle V, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{c(A)} & \langle K[A], \xi(c), \gamma_P, \gamma_O \cdot \gamma', \xi, \phi \uplus \mathrm{dom}(\gamma'), h_P, h_O \rangle \\
& \multicolumn{3}{l}{\text{when } c : \sigma, \gamma_P(c) = K, \text{ and } (A, \gamma') \in \mathbf{AVal}_\sigma(V)} \\
(OQ) & \langle K[fV], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle & \xrightarrow{f(A,c')} & \langle V'A, \gamma_P, \gamma_O \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \\
& & & \quad \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h_P, h_O \rangle \\
& \multicolumn{3}{l}{\text{when } f : \sigma \to \sigma', c' : \sigma', \gamma_P(f) = V' \text{ and } (A, \gamma') \in \mathbf{AVal}_\sigma(V)}
\end{array}
$$

Fig. 9. Composite LTS for HOSC[HOSC]

- $\gamma_P \cdot \gamma_O$ *is well-typed;*
- $c \in \phi$ *with* $c : \sigma \vdash M : \sigma$*;*
- $\mathrm{dom}(h_P) \cap \mathrm{dom}(h_O) = \varnothing$.

The composite LTS, defined on such composite configurations, is given in Figure 9. Up to choice of name, it is deterministic.

Definition 25. *Two valid* HOSC*-configurations* $\mathbf{C}_P, \mathbf{C}_O$ *are said to be* compatible *if one of the two is active and the other one is passive, and, without loss of generality, supposing that* $\mathbf{C}_P$ *is the active configuration* $\langle M, c, \gamma_P, \xi_P, \phi_P, h_P \rangle$ *and* $\mathbf{C}_O$ *the passive configuration* $\langle \gamma_P, \xi_O, \phi_O, h_O \rangle$*, then* $\phi_O = \phi_P \uplus \{\circ, \diamond\}$ *and the composite configuration* $\langle M, c, \gamma_P, \gamma_O, \xi_P \cdot \xi_O, \phi_O, h_P, h_O \rangle$*, written* $\mathbf{C}_P \mathbin{\wedge\!\!\wedge} \mathbf{C}_O$*, is valid.*

Lemma 23. *Taking* $\mathbf{D}$ *a valid composite configuration and* $\mathbf{D}'$ *a composite configuration s.t.* $\mathbf{D} \stackrel{\mathbf{a}}{\Rightarrow} \mathbf{D}'$*, then* $\mathbf{D}'$ *is valid.*

Lemma 24. *Taking* $\mathbf{C}_P, \mathbf{C}_O$ *two compatible configurations, for all composite configuration* $\mathbf{D}'$*, if* $(\mathbf{C}_P \mathbin{\wedge\!\!\wedge} \mathbf{C}_O) \stackrel{\mathbf{a}}{\Rightarrow} \mathbf{C}'$ *then there exists two compatible configurations* $\mathbf{C}'_P, \mathbf{C}'_O$ *s.t.:*

- $\mathbf{D}' = \mathbf{C}'_P \mathbin{\wedge\!\!\wedge} \mathbf{C}'_O$*;*
- $\mathbf{C}_P \stackrel{\mathbf{a}}{\Rightarrow} \mathbf{C}'_P$ *and* $\mathbf{C}_O \stackrel{\mathbf{a}^\perp}{\Longrightarrow} \mathbf{C}'_O$.

Proof. Without loss of generality, we suppose that $\mathbf{C}_P$ is the active configuration and $\mathbf{C}_O$ the passive one. So we write $\mathbf{C}_P$ as $\langle M, c, \gamma_P, \phi, h_P \rangle$ and $\mathbf{C}_O$ as $\langle \gamma_O, \phi, h_O \rangle$.

– If $\mathbf{a}$ is a Player Answer $\bar{c'}(A)$, then there exists V, h'_P s.t.

$$(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) \xrightarrow{\tau} \langle V, c', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O\rangle$$

so that $(M, c, h_P) \to (V, c', h'_P)$. Then there exists K, c'' s.t. $\gamma_O(c') = K, \xi(c') = c''$ and there exists σ, γ' s.t. $c' : \sigma$ and $(A, \gamma') \in \mathbf{AVal}_\sigma(V)$, so that $\mathbf{D}' = \langle K[A], c'', \gamma_P \cdot \gamma', \gamma_O, \phi \uplus \mathrm{dom}(\gamma'), h'_P, h_O\rangle$.
We then define $\mathbf{C}'_P$ as $\langle \gamma_P \cdot \gamma', \phi \uplus \mathrm{dom}(\gamma'), h'_P\rangle$ and $\mathbf{C}'_O$ as $\langle K[A], c'', \gamma_O, \phi \uplus \mathrm{dom}(\gamma'), h_O\rangle$. One easily check that:
  - $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations;
  - $\mathbf{D}' = \mathbf{C}'_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}'_O$;
  - $\mathbf{C}_P \xrightarrow{\tau} \langle V, c', \gamma_P, \phi, h'_P\rangle \xrightarrow{\bar{c'}(A)} \mathbf{C}'_P$;
  - $\mathbf{C}_O \xrightarrow{c'(A)} \mathbf{C}'_O$.

– If $\mathbf{a}$ is a Player Question $\bar{f}(A, c')$, then there exists K, V, c'', h'_P s.t.

$$(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) \xrightarrow{\tau} \langle K[f\ V], c'', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O\rangle$$

so that $(M, c, h_P) \to (K[f\ V], c'', h'_P)$. Then there exists V' s.t $\gamma_O(f) = V'$, and there exists σ, σ', γ' s.t. $f : \sigma \to \sigma'$, and $(A, \gamma') \in \mathbf{AVal}_\sigma(V)$, so that $\mathbf{D}' = \langle V'A, c', \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \gamma_O, \xi \cdot [c' \mapsto c''], \phi \uplus \mathrm{dom}(\gamma') \cdot \{c'\}, h'_P, h_O\rangle$.
We then define $\mathbf{C}'_P$ as $\langle \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c''], \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h'_P\rangle$ and $\mathbf{C}'_O$ as $\langle V'A, c', \gamma_O, \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h_O\rangle$. One easily check that:
  - $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations;
  - $\mathbf{D}' = \mathbf{C}'_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}'_O$;
  - $\mathbf{C}_P \xrightarrow{\tau} \langle K[f\ V], c'', \gamma_P, \phi, h'_P\rangle \xrightarrow{\bar{f}(A,c')} \mathbf{C}'_P$;
  - $\mathbf{C}_O \xrightarrow{\bar{f}(A,c')} \mathbf{C}'_O$.

Lemma 25. *Taking $\mathbf{C}_P, \mathbf{C}_O$ two compatible configurations, if*

– $\mathbf{C}_P \xRightarrow{\mathbf{a}} \mathbf{C}'_P$;

– $\mathbf{C}_O \xRightarrow{\mathbf{a}^\perp} \mathbf{C}'_O$;

then $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations and $(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) \xRightarrow{\mathbf{a}} (\mathbf{C}'_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}'_O)$.

Proof. Without loss of generality, we suppose that $\mathbf{C}_P$ is the active configuration and $\mathbf{C}_O$ the passive one. So we write $\mathbf{C}_P$ as $\langle M, c, \gamma_P, \phi, h_P\rangle$ and $\mathbf{C}_O$ as $\langle \gamma_O, \phi, h_O\rangle$.

– If $\mathbf{a}$ is a Player Answer $\bar{c'}(A)$, then there exists V, h'_P s.t. $\mathbf{C}_P \xrightarrow{\tau} \langle V, c', \gamma_P, \phi, h'_P\rangle$ so that $(M, c, h_P) \to (V, c', h'_P)$. Then:
  - there exists σ s.t. $c' : \sigma$, and γ', s.t. $(A, \gamma') \in \mathbf{AVal}_\sigma(V)$ so that $\mathbf{C}'_P = \langle \gamma_P \cdot \gamma', \phi \uplus \mathrm{dom}(\gamma'), h'_P\rangle$;
  - there exists K, c'' s.t. $\gamma_O(c') = K, \xi(c') = c''$ and $\mathbf{C}'_O = \langle K[A], c'', \gamma_O, \phi \uplus \mathrm{dom}(\gamma'), h_O\rangle$.

Then one easily checks that $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations, and:

$$\begin{aligned}(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) &\xrightarrow{\tau} \langle V, c', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O\rangle \\ &\xrightarrow{\bar{c'}(A)} \langle K[A], c'', \gamma_P \cdot \gamma', \gamma_O, \phi \uplus \mathrm{dom}(\gamma'), h'_P, h_O\rangle\end{aligned}$$

so that $\langle K[A], c'', \gamma_P \cdot \gamma', \gamma_O, \phi \uplus \mathrm{dom}(\gamma'), h'_P, h_O\rangle = \mathbf{C}'_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}'_O$.

– If $\mathbf{a}$ is a Player Question $\bar{f}(A, c')$, there exists K, V, c'', h'_P s.t. $\mathbf{C}_P \xrightarrow{\tau} \langle K[f\ V], c'', \gamma_P, \phi, h'_P\rangle$ so that $(M, c, h_P) \to (K[f\ V], c'', h'_P)$. Then:
  - there exists σ, σ' s.t. $f : \sigma \to \sigma'$, and V', γ', s.t. $\gamma_O(f) = V'$ and $(A, \gamma') \in \mathbf{AVal}_\sigma(V)$ so that $\mathbf{C}'_P = \langle \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c''], \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h'_P\rangle$;
  - there exists V' s.t. $\gamma_O(f) = V'$ and $\mathbf{C}'_O = \langle V'A, c', \gamma_O, \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h_O\rangle$.

  Then one easily checks that $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations, and:

$$\begin{aligned}(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) &\xrightarrow{\tau} \langle K[f\ V], c'', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O\rangle \\ &\xrightarrow{\bar{f}(A,c')} \langle V'A, c', \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \gamma_O, \xi \cdot [c' \mapsto c''], \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h'_P, h_O\rangle\end{aligned}$$

  so that $\langle K[A], c'', \gamma_P \cdot \gamma' \cdot [c' \mapsto K], \gamma_O, \xi \cdot [c' \mapsto c''], \phi \uplus \mathrm{dom}(\gamma') \uplus \{c'\}, h'_P \cdot h_O\rangle = \mathbf{C}'_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}'_O$.

Definition 26. *A composite configuration* $\mathbf{D}$ terminates *following a trace* $\mathtt{t}$*, written* $\mathbf{D} \Downarrow^{\mathtt{t}}_{ter}$*, when there exists a* final *composite configuration* $\mathbf{D}_f = \langle (), \circ, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *s.t.* $\mathbf{D} \stackrel{\mathtt{t}}{\Rightarrow} \mathbf{D}_f$*. We often omit the trace* $\mathtt{t}$ *and simply write* $\mathbf{D} \Downarrow_{ter}$*.*

Definition 27. *A composite configuration* $\mathbf{D}$ errors *following a trace* $\mathtt{t}$*, written* $\mathbf{D} \Downarrow^{\mathtt{t}}_{err}$*, when there exists a composite configuration* $\mathbf{D}_f = \langle K[err()], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *s.t.* $\mathbf{D} \stackrel{\mathtt{t}}{\Rightarrow} \mathbf{D}_f$*. We often omit the trace* $\mathtt{t}$ *and simply write* $\mathbf{D} \Downarrow_{err}$*.*

Lemma 26. *Taking* $\mathbf{C}_P, \mathbf{C}_O$ *two compatible configurations if* $(\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O) \Downarrow^{\mathtt{t}}_{ter}$ *then:*

– *if* $\mathbf{C}_P$ *is active and* $\mathbf{C}_O$ *passive,* $\mathtt{t}$ *is even-length;*
– *if* $\mathbf{C}_P$ *is passive and* $\mathbf{C}_O$ *active,* $\mathtt{t}$ *is odd-length.*

Proof. By induction on the length of $\mathtt{t}$:

– If $\mathtt{t} = \epsilon$, then $\mathbf{C}_P \mathbin{\wedge\!\!\!\wedge} \mathbf{C}_O$ can be written as $\langle (), \circ, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$. Writing ϕ_P for the name environment component of $\mathbf{C}_P$, and ϕ_O for the one of $\mathbf{C}_O$, then $\phi = \phi_O = \phi_P \uplus \{\circ\}$. So necessarily is the $\mathbf{C}_O$ active one.
– If $\mathtt{t} = \mathbf{a} \cdot \mathtt{t}'$, then we conclude using Lemma 24 and the induction hypothesis.

Definition 28. *Taking* $\mathbf{C}_P, \mathbf{C}_O$ *two compatible configurations, one write* $(\mathbf{C}_P | \mathbf{C}_O) \downarrow^{\mathtt{t}}_{\mathbf{y}}$*, with* $\mathbf{y} \in \{ter, err\}$*, when* $\mathtt{t} \in \mathbf{Tr}(\mathbf{C}_P)$ *and*

– *if* $\mathbf{y} = ter$ *then* $\mathtt{t}^\perp \cdot \bar{\circ}(()) \in \mathbf{Tr}(\mathbf{C}_O)$*;*
– *if* $\mathbf{y} = err$ *then* $\mathtt{t}^\perp \cdot \bar{\diamond}((), c) \in \mathbf{Tr}(\mathbf{C}_O)$ *for some* $c \in$ CNames*;*

Lemma 27. *Taking $\mathbf{C}_P, \mathbf{C}_O$ two compatible configurations and* $\mathtt{t}$ *a trace, then* $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\mathtt{t}}_{\mathbf{y}}$ *iff* $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\mathtt{t}}_{\mathbf{y}}$*, with* $\mathbf{y} \in \{ter, err\}$.

Proof. We first prove that if $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\mathtt{t}}_{\mathbf{y}}$ then $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\mathtt{t}}_{\mathbf{y}}$ by induction on the length of $\mathtt{t}$:

- if $\mathtt{t}$ is empty and $\mathbf{y} = ter$, then $\bar{o}(()) \in \mathbf{Tr}(\mathbf{C}_O)$, so there exists γ_O, ϕ, h_O s.t. $\mathbf{C}_O \xrightarrow{\tau} \langle (), \circ, \gamma_O, \phi, h_O \rangle$. Since $\mathbf{C}_O$ is an active configuration, $\mathbf{C}_P$ must be a passive configuration, that we write as $\langle \gamma_P, \phi, h_P \rangle$. Then $\mathbf{C}_P \curlywedge \mathbf{C}_O = \langle (), \circ, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$, so that indeed $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\epsilon}_{ter}$.
- if $\mathtt{t}$ is empty and $\mathbf{y} = err$, then $e\bar{r}r((), c) \in \mathbf{Tr}(\mathbf{C}_O)$, so there exists γ_O, ϕ, h_O s.t. $\mathbf{C}_O \xrightarrow{\tau} \langle K[err()], c, \gamma_O, \phi, h_O \rangle$. Since $\mathbf{C}_O$ is an active configuration, $\mathbf{C}_P$ must be a passive configuration, that we write as $\langle \gamma_P, \phi, h_P \rangle$. Then $\mathbf{C}_P \curlywedge \mathbf{C}_O = \langle K[err()], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$, so that indeed $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\epsilon}_{ter}$.
- if $\mathtt{t} = \mathbf{a} \cdot \mathtt{t}'$, then there exists two configurations $\mathbf{C}'_P, \mathbf{C}'_O$ s.t.:
  - $\mathbf{C}_P \xRightarrow{\mathbf{a}} \mathbf{C}'_P$;
  - $\mathbf{C}_O \xRightarrow{\mathbf{a}^{\perp}} \mathbf{C}'_O$;
  - $(\mathbf{C}'_P|\mathbf{C}'_O) \downarrow^{\mathtt{t}'}_{ter}$.

  From Lemma 25, we get that $\mathbf{C}'_P, \mathbf{C}'_O$ are two compatible configurations and $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \xRightarrow{\mathbf{a}} (\mathbf{C}'_P \curlywedge \mathbf{C}'_O)$. Using the induction hypothesis we get that $(\mathbf{C}'_P \curlywedge \mathbf{C}'_O) \Downarrow^{\mathtt{t}'}_{\mathbf{y}}$. So $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\mathtt{t}}_{\mathbf{y}}$.

We now prove that if $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \Downarrow^{\mathtt{t}}_{\mathbf{y}}$ then $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\mathtt{t}}_{\mathbf{y}}$, by induction on the length of $\mathtt{t}$:

- if $\mathtt{t}$ is empty and $\mathbf{y} = ter$, then $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \xrightarrow{\tau} \langle (), \circ, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$. So $\mathbf{C}_O \xrightarrow{\tau} \langle (), \circ, \gamma_O, \phi, h_O \rangle$ and $\mathbf{C}_P = \langle \gamma_P, \phi, h_P \rangle$. Thus $\mathbf{C}_O \xRightarrow{\bar{o}(())} \langle \gamma_O, \phi, h_O \rangle$, so $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\epsilon}_{ter}$.
- if $\mathtt{t}$ is empty and $\mathbf{y} = err$, then $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \xrightarrow{\tau} \langle K[err()], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$. So $\mathbf{C}_O \xrightarrow{\tau} \langle K[err()], c, \gamma_O, \phi, h_O \rangle$ and $\mathbf{C}_P = \langle \gamma_P, \phi, h_P \rangle$. Thus $\mathbf{C}_O \xRightarrow{e\bar{r}r((),c)} \langle \gamma_O, \phi, h_O \rangle$, so $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\epsilon}_{ter}$.
- if $\mathtt{t} = \mathbf{a} \cdot \mathtt{t}'$, then there exists a composite configuration $\mathbf{D}'$ s.t. $(\mathbf{C}_P \curlywedge \mathbf{C}_O) \xRightarrow{\mathbf{a}} \mathbf{D}'$ and $\mathbf{D}' \Downarrow^{\mathtt{t}'}_{ter}$. From Lemma 24, we get the existence of two compatible configurations $\mathbf{C}'_P, \mathbf{C}'_O$ s.t.:
  - $\mathbf{D}' = \mathbf{C}'_P \curlywedge \mathbf{C}'_O$;
  - $\mathbf{C}_P \xRightarrow{\mathbf{a}} \mathbf{C}'_P$;
  - $\mathbf{C}_O \xRightarrow{\mathbf{a}^{\perp}} \mathbf{C}'_O$.

  From $(\mathbf{C}'_P \curlywedge \mathbf{C}'_O) \Downarrow^{\mathtt{t}'}_{\mathbf{y}}$, we get from the induction hypothesis that $(\mathbf{C}'_P|\mathbf{C}'_O) \downarrow^{\mathtt{t}'}_{\mathbf{y}}$. So $(\mathbf{C}_P|\mathbf{C}_O) \downarrow^{\mathtt{t}}_{\mathbf{y}}$.

Definition 29. *Taking γ, ξ a valid environment and c, c' two continuation names s.t. $c \prec^*_{\xi} c'$, we define the evaluation context $K_{c,c'}$ as:*

– $K_{c,c} \triangleq \bullet$
– $K_{c,c'} \triangleq K_{c'',c'}[K]$, *when* $\gamma(c) = K$ *and* $\xi(c) = c''$.

We write K_c *for* $K_{c,\circ}$.

Definition 30. *To an environment* γ*, we associate an idempotent substitution* δ *defined as the relation:*

– $\delta^0 \triangleq \{(f, V) \mid f \in \mathrm{dom}(\gamma) \wedge \gamma(f) = V\} \cup \{(c, K) \mid c \in \mathrm{dom}(\gamma) \wedge \gamma(c) = K\}$
– $\delta^{i+1} \triangleq \{(f, V\{\delta^i\}) \mid (f, V) \in \delta^i\} \cup \{(c, K\{\delta^i\}) \mid (c, K) \in \delta^i\}$ *where we write* $V\{\delta^i\}$ *for the action of the substitution* δ^i *to* V

then there exists $n \in \mathbb{N}$ *s.t.* $\delta^{n+1} = \delta^n$*, and* δ *is then defined as* δ^n.

One need this iterative construction to get the idempotency result, that corresponds to the fact that the support of the values and evaluation contexts in the codomain of δ are empty (i.e. they do not have continuation or functional names anymore). This is possible because there is no cycles between names.

Lemma 28. *Taking* $\mathbf{D} = \langle K[f\ V], c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *a valid composite configuration that is going to perform a question, with* $f \in \mathrm{dom}(\gamma)$*, where* $\gamma = \gamma_P \cdot \gamma_O$*, there exists a functional name* g*, an abstract value* A*, a composite configuration* $\mathbf{D}'$ *and a trace* $\mathtt{t}$ *formed by questions s.t.:*

– $\gamma(g)$ *is a* λ*-abstraction* $\lambda x.M$*;*
– $\delta(f) = \delta(g)$*, writing* δ *for the idempotent substitution associated to* γ*;*
– $\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}'$*;*
– $\mathbf{D}'$ *can be written as* $\langle g\ A, c', \gamma_P \cdot \gamma'_P, \gamma_O \cdot \gamma'_O, \phi \uplus \mathrm{dom}(\gamma'_P), h_P, h_O\rangle$*;*
– $A\{\delta'\} = V$*, with* δ' *the idempotent substitution associated to* $\gamma'_P \cdot \gamma'_O$*;*
– $K^\gamma_{c',c} = \bullet$.

Lemma 29. *Let* $\mathbf{D} = \langle V, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$ *be a valid composite configuration that is going to perform an answer. Suppose that there exists* c' *s.t.* $c \prec^*_\gamma c'$ *and* $K^\gamma_{c,c'} = \bullet$*. Then there exists a composite configuration* $\mathbf{D}' = \langle A, c', \gamma_P \cdot \gamma'_P, \gamma_O \cdot \gamma'_O, \phi \uplus \mathrm{dom}(\gamma'_P), h_P, h_O\rangle$ *and a trace* $\mathtt{t}$ *formed only by answers s.t.* $\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}'$ *and* $A\{\delta'\} = V$*, with* δ' *the idempotent substitution associated to* $\gamma'_P \cdot \gamma'_O$.

Definition 31. *One define the configuration transformation* θ *from valid composite configurations to pair formed by a term and a heap, defined as*

$$\theta : \langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle \mapsto ((K^\gamma_c[M])\{\delta\}, (h_P \cdot h_O)\{\delta\})$$

writing γ *for* $\gamma_P \cdot \gamma_O$ *and* δ *for the idempotent substitution associated to* γ.

Lemma 30. *Taking* $\mathbf{D}, \mathbf{D}'$ *two valid composite configuration and* $\mathbf{a}$ *an action (different of* τ*) s.t.* $\mathbf{D} \xrightarrow{\mathbf{a}} \mathbf{D}'$ *then* $\theta(\mathbf{D}) = \theta(\mathbf{D}')$.

Proof. Let us write $\mathbf{D}$ as $\langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O\rangle$. Without loss of generality, we suppose the composite configuration $\mathbf{D}$ to be P-active, i.e. $c \in \mathrm{dom}(\gamma_O)$

We reason by case analysis over α:

- If $\alpha = \bar{c}(A)$, so that M is a value V. Then we have:
  - $\gamma_O(c) = K$ and $c : \tau$ for some context K and type τ;
  - $\gamma'_O = \gamma_O$, $\gamma'_P = \gamma_P \cdot \gamma_A$ and $\phi' = \phi \uplus \mathrm{dom}(\gamma_A)$; with $(A, \gamma_A) \in \mathbf{AVal}_\tau(V)$;
  - $h'_P = h_P$ and $h'_O = h_O$;
  - $M' = K[A]$.

  We conclude using these and the fact that:
  - $K^\gamma_c = K^\gamma_{c'}[K]$, where $c' = \xi(c)$;
  - $A\{\gamma_A\} = V$;

  that $(K^\gamma_c[V])\{\delta\} = (K^{\gamma'}_{c'}[K[A]])\{\delta'\}$. So $\theta(\mathbf{D}) = \theta(\mathbf{D}')$.
- If $\alpha = \bar{f}(A, c')$, so that M is a callback $K[f\ V]$ for some context K, value V, and functional name f. Then we have:
  - $\gamma_O(f) = V'$ and $f : \sigma \to \sigma'$ for some value V and type σ, σ';
  - $\gamma'_O = \gamma_O$, $\gamma'_P = \gamma_P \cdot \gamma_A \cdot [c' \mapsto K]$, $\xi' = \xi \cdot [c' \mapsto c]$ and $\phi' = \phi \uplus \mathrm{dom}(\gamma_A) \cdot \{c'\}$, with $(A, \gamma_A) \in \mathbf{AVal}_\sigma(V)$;
  - $h'_P = h_P$ and $h'_O = h_O$;
  - $M' = V'\ A$.

  We conclude using these and the fact that:
  - $K^\gamma_{c'} = K^\gamma_c[K]$;
  - $\gamma_O(f) = V'$;
  - $A\{\gamma_A\} = V$;

  that $(K^\gamma_c[K[f\ V]])\{\delta\} = (K^{\gamma'}_{c'}[V' A])\{\delta'\}$. So $\theta(\mathbf{D}) = \theta(\mathbf{D}')$.

Definition 32. *Taking $\mathbf{D}, \mathbf{D}'$ two composite configuration, we write $\mathbf{D} \rightsquigarrow \mathbf{D}'$ when there exists a trace $\mathtt{t}$ of actions (without any τ-actions) s.t. $\mathbf{D} \xrightarrow{\mathtt{t} \cdot \tau} \mathbf{D}'$.*

Lemma 31. *The configuration transformation θ is a functional bisimulation between the transition system over composite configurations $(CompConf, \rightsquigarrow)$ and the operational transition system $(\Lambda \times \mathrm{Heap}, \to)$, that is, for all valid composite configuration $\mathbf{D}$:*

- *for all composite configuration $\mathbf{D}'$, if $\mathbf{D} \rightsquigarrow \mathbf{D}'$ then $\theta(\mathbf{D}) \to \theta(\mathbf{D}')$;*
- *for all pairs (N, h) formed by a term an a heap h', if $\theta(\mathbf{D}) \to (N, h')$ then there exists a valid composite configuration $\mathbf{D}'$ s.t. $\mathbf{D} \rightsquigarrow \mathbf{D}'$ and $(N, h') = \theta(\mathbf{D}')$*

Proof. We write:

- $\mathbf{D}$ as $\langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$;
- γ for $\gamma_P \cdot \gamma_O$;
- δ for the idempotent substitution associated to $\gamma_P \cdot \gamma_O$;
- $\theta(\mathbf{D})$ as $(K^\gamma_c[M])\{\delta\}, h)$ with $h = (h_P \cdot h_O)\{\delta\}$.

We first suppose that $\mathbf{D} \rightsquigarrow \mathbf{D}'$, i.e. there exists a trace $\mathtt{t}$ of actions (without any τ) and a composite configurations $\mathbf{D}_1$ s.t. $\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}_1 \xrightarrow{\tau} \mathbf{D}'$. From Lemma 30, we get that $\theta(\mathbf{D}) = \theta(\mathbf{D}_1)$.

Without loss of generality, we suppose the composite configuration $\mathbf{D}_1$ is P-active. We write $\mathbf{D}'$ as $\langle M', c', \gamma'_P, \gamma'_O, \phi', h'_P, h_O \rangle$ and $\mathbf{D}_1$ as $\langle M_1, c_1, \gamma'_P, \gamma'_O, \phi', h_P, h_O \rangle$, so that we have $(M_1, c_1, h_P) \to (M', c', h'_P)$.

From Lemma 9, writing δ' for the idempotent substitution associated to $\gamma'_P \cdot \gamma'_O$, and δ'_C for its restriction to the domain of continuation names, one has that $(K^{\gamma'}_{c_1}[M]\{\delta'_C\}, h_P\{\delta'_C\}) \to (K^{\gamma'}_{c'}[M']\{\delta'_C\}, h'_P\{\delta'_C\})$. Extending the heap with h_O and the substitution to δ', we get that $(K^{\gamma'}_{c_1}[M]\{\delta'\}, h) \to (K^{\gamma'}_{c'}[M']\{\delta'\}, (h'_P \cdot h_O)\{\delta'\})$, i.e. $\theta(\mathbf{D}_1) \to \theta(\mathbf{D}')$.

Now, we suppose that there exists a term N and a heap h' s.t. $\theta(\mathbf{D}) \to (N, h')$. From Lemma 20, there is three possible cases for the reduction $\theta(\mathbf{D}) \to (N, h')$:

- Either $(M, c, h_P \cdot h_O)$ is reducible. Without loss of generality, we suppose the composite configuration $\mathbf{D}$ is P-active, so that (M, c, h_P) is reducible. Then there exists (M', c', h'_P) s.t.:
  - $(M, c, h_P) \to (M', c', h'_P)$;
  - $N = (K^{\gamma'}_{c'}[M'])\{\delta\}$;
  - $h' = (h'_P \cdot h_O)\{\delta\}$.

  So we take $\mathbf{D'} = \langle M', c', \gamma_P, \gamma_O, \xi, \phi, h'_P, h_O\rangle$ so that $\mathbf{D} \xrightarrow{\tau} \mathbf{D'}$.
- Or M is a callback:
  - $M = K[f\ V]$ for some context K, value V, and functional name f;
  - $\delta(f)$ is a λ-abstraction that we write $\lambda x.P$ (with $x \notin \mathrm{dom}(\delta)$);
  - $N = (K^{\gamma}_{c}[K[P\{V/x\}]])\{\delta\}$;
  - $h' = h$;

  From Lemma 28, there exists a functional name g, an abstract value A_1, a composite configuration $\mathbf{D}_1$ and a trace $\mathtt{t}$ formed by questions s.t.:
  - $\gamma(g)$ is a λ-abstraction $\lambda x.\hat{P}$;
  - $\delta(f) = \delta(g)$;
  - $\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}_1$;
  - $\mathbf{D}_1$ can be written as $\langle g\ A_1, c_1, \gamma_P \cdot \gamma_{1,P}, \gamma_O \cdot \gamma_{1,O}, \phi \uplus \mathrm{dom}(\gamma_{1,P}), h_P, h_O\rangle$;
  - $A_1\{\delta_1\} = V$, with δ_1 the idempotent substitution associated to $\gamma_{1,P} \cdot \gamma_{1,O}$;
  - $K^{\gamma_1}_{c_1,c} = K$.

  Without loss of generality, we suppose the composite configuration $\mathbf{D}_1$ is P-active. Then we have:

  $$\begin{array}{rl} \mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}_1 \xrightarrow{\bar{g}(A_2,c_2)} & \overbrace{\langle(\lambda x.\hat{P})\ A_2, c_2, \gamma_{2,P}, \gamma_O \cdot \gamma_{1,O}, \phi_2, h_P, h_O\rangle}^{\mathbf{D}_2} \\ \xrightarrow{\tau} & \underbrace{\langle \hat{P}\{A_2/x\}, c_2, \gamma_{2,P}, \gamma_O \cdot \gamma_{1,O}, \phi_2, h_P, h_O\rangle}_{\mathbf{D'}} \end{array}$$

  with $\gamma_{2,P} = \gamma_P \cdot \gamma_{1,P} \cdot \gamma_{A_2} \cdot [c_2 \mapsto (\bullet, c_1)]$ and $A_2\{\gamma_{A_2}\} = A_1$. From Lemma 30, we have that $\theta(\mathbf{D}) = \theta(\mathbf{D}_2)$.
  We prove that $(\hat{P}\{A_2/x\})\{\delta_2\} = P\{V\{\delta\}/x\}$ from the fact that:
  - $A_2\{\delta_2\} = V\{\delta\}$ since $A_1\{\delta_1\} = V$ and $A_1 = A_2\{\gamma_{A_2}\}$;
  - $\hat{P}\{\delta\} = P$ since $\delta(f) = \delta(g)$, $\delta(f) = \lambda x.P$ and $\gamma(g) = \lambda x.\hat{P}$.

  Finally, from $K^{\gamma_1}_{c_1,c} = K$ and $\gamma_2(c_2) = (\bullet, c_1)$, we get that $K^{\gamma_2}_{c_2} = K^{\gamma}_{c}[K]$. So $\theta(\mathbf{D'}) = (N, h)$.

– Or M is a value V and K_c^γ an evaluation context larger than $\bullet$. Then there exists a continuation name c_1 s.t.:
  - $c \prec_\gamma^* c_1$
  - $K_{c,c_1}^\gamma = \bullet$.
  - $\gamma(c_1) = K$ with K an evaluation context larger than $\bullet$;

  From Lemma 29, there exists an abstract value A_1, a composite configuration $\mathbf{D}_1$ and a trace $\mathtt{t}$ formed by answers s.t.:
  - $\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}_1$;
  - $\mathbf{D}_1$ can be written as $\langle A_1, c_1, \gamma_P \cdot \gamma_{1,P}, \gamma_O \cdot \gamma_{1,O}, \phi \uplus \mathrm{dom}(\gamma_{1,O}), h_P, h_O \rangle$;
  - $A_1\{\delta_1\} = V$, with δ_1 the idempotent substitution associated to $\gamma_{1,P} \cdot \gamma_{1,O}$;

  Without loss of generality, we suppose the composite configuration $\mathbf{D}_1$ is P-active. Then we have:

$$\mathbf{D} \xrightarrow{\mathtt{t}} \mathbf{D}_1 \xrightarrow{\bar{c_1}(A_2)} \overbrace{\langle K[A_2], c_2, \gamma_{2,P}, \gamma_{2,O}, \phi_2, h_P, h_O \rangle}^{\mathbf{D}_2}$$

  with $\xi(c_1) = c_2$,$\gamma_{2,P} = \gamma_{1,P} \cdot \gamma_{A_2}$ and $A_2\{\gamma_{A_2}\} = A_2$.
  From Lemma 30, we have that $\theta(\mathbf{D}) = \theta(\mathbf{D}_2)$. From $K_{c,c_1}^\gamma = \bullet$, we get that $K_c^\gamma = K_{c_1}^{\gamma_1}$, so that $K_{c_2}^{\gamma_2}[K] = K_c^\gamma$. Since K is larger than $\bullet$, $K[A_2]$ cannot be a value, so from Lemma 19 we have that:
  - either $(K[A_2], c_2, h_P)$ is reducible, and we conclude using a similar reasoning as in the first case, on $\mathbf{D}_2$.
  - or $K[A_2]$ is a callback, and we conclude using a similar reasoning as in the second case, on $\mathbf{D}_2$.

Corollary 5. *Taking* $\mathbf{D}$ *a valid composite configurations,* $\mathbf{D} \Downarrow_{ter}$ *iff* $\theta(\mathbf{D}) \Downarrow_{ter}$.

Proof. We write $\mathbf{D}$ as $\langle M, c, \gamma_P, \gamma_O, \xi, \phi, h_P, h_O \rangle$.

We first prove that if $\mathbf{D} \Downarrow_{ter}$ then $\theta(\mathbf{D}) \Downarrow_{ter}$. From $\mathbf{D} \Downarrow_{ter}$, we get the existence of a sequence of reductions $\mathbf{D} \rightsquigarrow^* \overbrace{\langle (), \circ, \gamma_{f,P}, \gamma_{f,O}, \phi_f, h_{f,P}, h_{f,O} \rangle}^{\mathbf{D}_f}$. We reason by induction over the length of this reduction.

– if $\mathbf{D} = \mathbf{D}_f$, then $\theta(\mathbf{D}) = ((), \_)$ since $M = ()$ and $c = \circ$ so that $K_c^{\gamma_P \cdot \gamma_O} = \bullet$.
– if there exists a composite configuration $\mathbf{D}'$ s.t. $\mathbf{D} \rightsquigarrow \mathbf{D}' \rightsquigarrow^* \mathbf{D}_f$, then by induction hypothesis $\theta(\mathbf{D}') \Downarrow_{ter}$, and from Theorem 31 one has that $\theta(\mathbf{D}) \to \theta(\mathbf{D}')$, so that $\theta(\mathbf{D}) \Downarrow_{ter}$.

We now prove that if $\theta(\mathbf{D}) \Downarrow_{ter}$ then $\mathbf{D} \Downarrow_{ter}$. From $\theta(\mathbf{D}) \Downarrow_{ter}$ we get the existence of $((), h)$ s.t. $\theta(\mathbf{D}) \to^* ((), h)$. We reason by induction over the length of this reduction.

– if the reduction is empty, then $\theta(\mathbf{D}) = ((), h)$. So necessarily $M = ()$ and $K_{c,\circ}^\gamma = \bullet$. Then from Lemma 29, $\theta(\mathbf{D}) \Downarrow_{ter}$.
– if there exists (M', h') s.t. $\theta(\mathbf{D}) \to (M', h') \to^* ((), h)$, then from Theorem 31, there exists a configuration $\mathbf{D}'$ s.t. $\theta(\mathbf{D}) \to \theta(\mathbf{D}')$ and $\theta(\mathbf{D}') = (M', h')$. Then by induction hypothesis, since $\theta(\mathbf{D}') \to^* ((), h)$, we get that $\theta(\mathbf{D}') \Downarrow_{ter}$, so that $\theta(\mathbf{D}) \Downarrow_{ter}$.

Corollary 6. *Taking* $\mathbf{D}$ *a valid composite configurations,* $\mathbf{D} \Downarrow_{err}$ *iff* $\theta(\mathbf{D}) \Downarrow_{err}$.

Finally, we can prove Lemma 4

Lemma 32 (Correctness). *Let* $\Gamma \vdash M : \tau$ *be a cr-free* HOSC *term, let* Σ, h, K, γ *be as above,* $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$*, and* $c : \tau$ $(c \notin \circ)$*. Then*

- $(K[M\{\gamma\}], h) \Downarrow_{err}$ *iff there exist* t, c' *such that* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_M^{\rho_{\vec{A_i}}, c})$ *and* $t^\perp \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}, c})$.
- $(K[M\{\gamma\}], h) \Downarrow_{ter}$ *iff there exist* t, A, τ *such that* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_M^{\rho_{\vec{A_i}}, c})$ *and* $t^\perp \circ^-_{\tau'}(A) \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}, c})$.

Moreover, t *must satisfy* $\nu(t) \cap (\circ \cup \{\diamond\}) = \emptyset$.

Proof. Let $\mathbf{y} \in \{ter, err\}$. Note that $(K[M\{\gamma\}], h) \Downarrow_{\mathbf{y}}$ iff $\theta(\mathsf{C}_M^{\rho_{\vec{A_i}}, c} \mathbin{\bar{\wedge}} \mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}}) \Downarrow_{\mathbf{y}}$. From Corollary 5 and 6, this is equivalent to the existence of a trace $\mathtt{t}$ such that $(\mathsf{C}_M^{\rho_{\vec{A_i}}, c} \mathbin{\bar{\wedge}} \mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}}) \Downarrow_{\mathbf{y}}^{\mathtt{t}}$. By Lemma 27, this is the same as $(\mathsf{C}_M^{\rho_{\vec{A_i}}, c} | \mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}}) \downarrow_{\mathbf{y}}^{\mathtt{t}}$, which implies the Lemma.

D Additional material for Section 4 (GOSC[HOSC])

D.1 Proof of Lemma 7 (visibility)

We write $\mathbf{C} \xrightarrow{t} \mathbf{C}'$ to say that there exists a sequence of transitions from $\mathbf{C}$ to $\mathbf{C}'$ such that the collected labels, including τ transitions, give a trace t. The proof is based on an auxiliary lemma (Lemma 33), which generalizes P-visibility to configurations, enabling an inductive proof.

Lemma (Original Statement of Lemma 7). Let $\mathbf{C}_O = \mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}, c}$, where h, K, γ are from GOSC, and $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$. All traces in $\mathbf{Tr}_{\mathrm{HOSC}}^{\mathrm{even}}(\mathbf{C}_O)$ are P-visible.

Proof. Suppose $\mathbf{C}_O \xRightarrow{a_1 \cdots a_{2i+1}} \mathbf{C}$ and $\mathbf{C} \xrightarrow{\tau^*} \mathbf{C}' \xrightarrow{a_{2i+2}} \mathbf{C}''$. By Lemma 33, $\mathbf{C}' = \langle M', c', \cdots \rangle$ with $\nu(M', c') \subseteq \mathsf{Vis}_P(a_1 \cdots a_{2i+1})$. Because the O-names in a_{2i+2} come from $\nu(M', c')$, P-visibility follows.

Lemma 33. *Suppose* $\mathbf{C}_O \xrightarrow{a_1 \cdots a_k} \mathbf{C}$.

1. *If* $\mathbf{C} = \langle \gamma, \xi, \phi, h \rangle$ *then, for any* $n \in \mathrm{dom}(\gamma)$*, if* n *was introduced in* a_{2i} $(0 \leq i \leq k/2)$ *then* $\nu(\gamma(n)) \subseteq \mathsf{Vis}_P(a_1 \cdots a_{2i-1})$ *and if* $n \in$ CNames *then* $\xi(n) \in \mathsf{Vis}_P(a_1 \cdots a_{2i-1})$ *(introduced in* a_0 *is taken to mean* $\diamond, \circ$ *and* $\mathsf{Vis}_P(a_1 \cdots a_0)$ *stands for* $\{\diamond, \circ\}$*).*
2. *If* $\mathbf{C} = \langle M, c, \gamma, \xi, \phi, h \rangle$ *then* $\nu(M, c) \subseteq \mathsf{Vis}_P(a_1 \cdots a_k)$ *and all of the conditions listed above hold.*

Proof. By induction on the number of transitions between $\mathbf{C}_O$ and $\mathbf{C}$, including τ-transitions.

The base case is $\mathbf{C}_O = \mathbf{C}$. The Lemma then holds because $\nu(\gamma) \subseteq \{\diamond\}$, $\xi(c) = \circ$, and $\mathsf{Vis}_P(a_1 \cdots a_0) = \{\diamond, \circ\}$.

Suppose $\mathbf{C}_O \xrightarrow{a_1 \cdots a_k} \mathbf{C}'$ and $\mathbf{C}_O \xrightarrow{t} \mathbf{C} \xrightarrow{x} \mathbf{C}'$, where t is a trace and x is an action or $x = \tau$.

- If $x = \tau$ then γ, ξ do not change during the transition and the reduction does not generate new names by Lemma 6. Hence, the Lemma follows from IH.
- Suppose x is an O-action, i.e. $x = a_k$. Then $\mathbf{C}' = \langle M', c', \gamma', \xi', \phi', h' \rangle$ and $\mathbf{C} = \langle \gamma', \xi', \phi' \setminus A, h' \rangle$. By IH for $\mathbf{C}$, all the conditions for γ', ξ' hold, so it remains to check $\nu(M', c')$.
  - If $x = c''(A'')$ then $\nu(M', c') = \nu(\gamma'(c'')[A''], \xi'(c''))$. By IH for $\mathbf{C}, c''$, assuming c'' was introduced in a_{2i}, we get $\nu(M', c') \subseteq \mathsf{Vis}_P(a_1 \cdots a_{2i-1}) \cup \nu(A'') = \mathsf{Vis}_P(a_1 \cdots a_k)$.
  - If $x = f(A'', c'')$ then $\nu(M', c') = \nu(\gamma'(f)[A''], c'')$. By IH for $\mathbf{C}, f$, assuming f was introduced in a_{2i}, we get $\nu(M', c') \subseteq \mathsf{Vis}_P(a_1 \cdots a_{2i-1}) \cup \nu(A'') \cup \{c''\} = \mathsf{Vis}_P(a_1 \cdots a_k)$.
- Suppose x is a P-action, i.e. $x = a_k$. Then $\mathbf{C}' = \langle \gamma', \xi', \phi', h' \rangle$.
  - If $x = \bar{c''}(A'')$ then $\mathbf{C} = \langle V, c'', \gamma' \setminus \nu(A''), \xi', \phi \setminus \nu(A''), h' \rangle$. By IH, $\gamma' \setminus \nu(A'')$ and ξ' satisfy the Lemma. It suffices to check $\gamma'(n)$ for $n \in \nu(A'')$. Observe that then $\nu(\gamma'(n)) \subseteq \nu(V, c'')$ and, by IH for $\mathbf{C}$, $\nu(V, c'') \subseteq \mathsf{Vis}_P(a_1 \cdots a_{k-1})$, as required.
  - If $x = \bar{f}(A'', c'')$ then $\mathbf{C} = \langle K[fV], c''', \gamma' \setminus X, \xi' \setminus \{c''\}, \phi \setminus X, h' \rangle$, where $X = \nu(A'') \cup \{c''\}$. By IH, $\gamma' \setminus X$ and $\xi' \setminus \{c''\}$ satisfy the Lemma. It suffices to check $\gamma'(n)$ for $n \in \nu(A'')$, $\gamma'(c'')$ and $\xi'(c'')$. Observe that then $\nu(\gamma'(n)) \cup \nu(\gamma'(c'')) \cup \{\xi'(c'')\} \subseteq \nu(K[fV], c''')$ and, by IH for $\mathbf{C}$, $\nu(K[fV], c''') \subseteq \mathsf{Vis}_P(a_1 \cdots a_{k-1})$, as required.

D.2 Proof of Theorem 3

Proof. Suppose $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$. Consider Σ, h, K, γ (as in the definition of $\lesssim_{err}^{\mathrm{GOSC}(ciu)}$) such that $(K[M_1\{\gamma\}], h) \Downarrow_{err}$. In particular, h, K, γ consist of GOSC syntax. Suppose $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$ and $c : \tau$ ($c \notin \circ$). By Lemma 4 (left-to-right), there exist t, c' such that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{M_1}^{\rho_{\vec{A_i}}, c})$ and $t^\perp \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}, c})$. By Lemma 7, $t^\perp \bar{\diamond}((), c')$ is P-visible. Thus, t is O-visible and, by Lemma 36 (right-to-left), $t \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M_1}^{\rho_{\vec{A_i}}, c})$. From $\mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{GOSC}}(\Gamma \vdash M_2)$, we get $t \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M_2}^{\rho_{\vec{A_i}}, c})$. By Lemma 36 (left-to-right), $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{M_2}^{\rho_{\vec{A_i}}, c})$. Because $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{M_2}^{\rho_{\vec{A_i}}, c})$ and $t^\perp \bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_{h,K,\gamma}^{\vec{\gamma_i}, c})$, by Lemma 4 (right-to-left), we can conclude $(K[M_2\{\gamma\}], h) \Downarrow_{err}$. Thus, $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}(ciu)} M_2$.

D.3 Proof of Lemma 9

Lemma 5 follows from the lemma given below for $i = 0$. Consider $h' = h_0$, $K' = \gamma_0(c)$, $\gamma' = \gamma_0 \setminus c$. We have $\nu(\mathrm{img}(\gamma_0), \mathrm{img}(h_0)) \subseteq \circ \uplus \{\diamond\}$. As names $\circ_\sigma$ can only occur inside terms of the form $\mathrm{cont}\,(K', \circ_\sigma)$, we can conclude that $(h', K', \gamma') = (h_\circ, K_\circ, \gamma_\circ)$, where h, K, γ are from GOSC.

Lemma 34. *Suppose* $\phi \uplus \{\diamond\} \subseteq \mathrm{FNames}$, $c \in \mathrm{CNames}$ *and* $t = o_1 p_1 \cdots o_n p_n$ *is a P-visible* $(\circ \uplus \{\diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. Given* $0 \le i \le n$, *let* $t_i = o_{i+1} p_{i+1} \cdots o_n p_n$. *There exist passive configurations* $\mathbf{C}_i$ *such that* $\mathbf{Tr}^{\mathrm{even}}(\mathbf{C}_i)$ *consists of even-length prefixes of* $o_{i+1} p_{i+1} \cdots o_n p_n$ *(along with their renamings via permutations on* Names *that fix* ϕ_i*). Moreover,* $\mathbf{C}_i = \langle \gamma_i, \xi_i, \phi_i, h_i \rangle$ $(0 \le i \le n)$*, where*

- $\mathrm{dom}(\gamma_i)$ *consists of* $\phi \cup \{c\}$ *and all names introduced by P in* $o_1 p_1 \cdots o_i p_i$*;*
- $\mathrm{img}(\gamma_i)$ *contains GOSC syntax;*
- $\nu(\gamma_i(x)) \subseteq \mathsf{Vis}_P(o_1 p_1 \cdots o_i)$ *if* x *has been introduced in* p_i *(*$\phi \uplus \{c\}$ *are deemed to have been introduced in* p_0 *and we assume* $\mathsf{Vis}_P(o_1 \cdots o_0) = \circ \uplus \{\diamond\}$*);*
- *for all* $d \in \mathrm{dom}(\gamma_i) \cap \mathrm{CNames}$*, if* $d : \sigma_d$ *and* d *was introduced in* p_j *then* $\vdash \gamma_i(d) : \sigma_d \to \sigma_j$*, where* $top_P(o_1 \cdots o_j) : \sigma_j$*;*
- $\mathrm{dom}(\xi_i)$ *consists of* c *and all continuation names introduced by P in* $o_1 p_1 \cdots o_i p_i$*;*
- *for all* $d \in \mathrm{dom}(\xi_i)$*,* $\xi_i(d) = top_P(o_1 \cdots o_j)$ *if* d *was introduced in* p_j *(we regard* c *as being introduced in* p_0 *and define* $top_P(o_1 \cdots o_0) = \circ_{\tau'}$*);*
- ϕ_i *consists of* $\circ \uplus \{\diamond\} \uplus \phi \uplus \{c\}$ *and all names introduced in* $o_1 p_1 \cdots o_i p_i$*;*
- *for all* $0 \le i \le n$*,* $h_i = \{time \mapsto i\}$*, where* $time : \mathrm{ref\,Int}$.

Proof. Note that the heap will consist of a single reference only, which will correspond to counting steps in the translation. At every step of the translation, the value of the reference will be used to schedule the right actions and disable others.

The above description already specifies ϕ_i, $\mathrm{dom}(\gamma_i)$, ξ_i and h_i. To complete the definition of $\mathbf{C}_i$, it remains to specify the environments γ_i. Recall that, we need to define $\gamma_0(x)$ for $x \in \phi \cup \{c_P\}$ and, in other cases, $\gamma_j(x)$ $(x \in \mathrm{Names})$ will be defined for all $j \ge i$ if x was introduced by P in p_i. Recall also that once $\gamma_j(x)$ is defined, it never changes. Hence if x was introduced by in p_i, we will only specify $\gamma_i(x)$ on the understanding that $\gamma_{i'}(x) = \gamma_i(x)$ for all $i' > i$.

We define $\gamma_i(x)$ by induction using the reverse order of name introduction in t, i.e. when defining $\gamma_i(x)$ we will refer to $\gamma_{i'}(y)$, where y is introduced in a later move in t. In particular, the names $\phi \cup \{c\}$ are deemed to be introduced first. Once $\gamma_i(x)$ is defined, we will argue that $\nu(\gamma_i(x)) \subseteq \mathsf{Vis}_P(o_1 p_1 \cdots o_i)$.

- Suppose $f : \sigma_f \to \tau_f$ is a function name introduced by P in action p_i $(1 \le i \le n)$ or $f \in \phi$, in which case we let $i = 0$. Consider all subsequent occurrences of f in t: suppose $\mathcal{I}_f = \{i < u \le n \mid o_u = f(A_u, c_u)\}$, i.e. $\mathcal{I}_f$ contains all the time points when it is necessary to respond to $f(A', c')$. Then we let

  $$\gamma_i(f) = \lambda x.\,(time := !time{+}1); \mathrm{if}\ (!time \in \mathcal{I}_f)\ (assert(x \sim A_{!time}); M_{!time})\ \Omega,$$

where $(assert(x \sim A_{!time}); M_{!time})$ is shorthand for code that performs case distinction on $!time$ and directs reduction to $(assert(x \sim A_u); M_u)$ for $u = !time \in \mathcal{I}_f$. The term $assert(x \sim A_u)$ has been defined earlier, so we specify M_u $(u \in \mathcal{I}_f)$, aiming to have $x : \sigma_f \vdash M_u : \tau_f$ in each case. M_u will depend on the shape of p_u. Note that if $\mathcal{I}_f = \emptyset$, i.e. f is not used in t, then the construction degenerates to $\gamma_i(f) = \lambda x.\,(time :=!time + 1); \Omega$.

$p_u = \bar{c'_u}(A'_u)$ As $u > i$, γ_u is already defined for all names in A'_u. Let $V = A'_u\{\gamma_u\}$. Recall that $o_u = f(A_u, c_u)$. We let

$$M_u = \text{call/cc}(\ y.\ (\text{throw V to cont}\,(\bullet, c'_u))\ [y/\text{cont}\,(\bullet, c_u)]\ [\pi x/A_u]\ ).$$

- $[y/\text{cont}\,(\bullet, c_u)]$ is meant to mimic the reversal of the reduction rule for call/cc: because after o_u the continuation name in the active configuration will be c_u, the then current continuation will be $\text{cont}\,(\bullet, c_u)$. Since all continuation names c' are only ever used via the term $\text{cont}\,(\bullet, c')$, the substitution $[y/\text{cont}\,(\bullet, c_u)]$ will remove all occurrences of c_u from V.
- The substitution $[\pi x/A_u]$ has been defined before the first definability proof.

Note that, because of throw, M_u can indeed be given type τ_f. Overall, the shape of M_u guarantees the desired progression $(o_u p_u)$ at time u (the configuration will reduce to $(V, c'_u, \cdots)$, to be followed by $p_u = \bar{c'_u}(A'_u)$). Because we can assume $\nu(\gamma_u(x)) \subseteq \mathsf{Vis}_P(o_1 \cdots o_u)$ for any x introduced in p_u (IH), we have $\nu(V) = \nu(A'_u\{\gamma_u\}) \subseteq \mathsf{Vis}_P(o_1 \cdots o_u)$. As all names introduced in o_u will be substituted for, we have $\nu(M_u) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i) \cup \{c'_u\}$. However, by P-visibility, we have $c'_u \in \mathsf{Vis}_P(o_1 \cdots o_u)$, so either $c'_u = c_u$ or $c'_u \in \mathsf{Vis}_P(o_1 \cdots o_i)$. Either way, we can conclude $\nu(M_u) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i)$, i.e. $\nu(\gamma_i(f)) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i)$.

$p_u = \bar{f'}(A'_u, c'_u)$ As in the previous case, by IH, γ_u is already defined for all names in A'_u and c'_u. Let $V = A'_u\{\gamma_u\}$ and $K = \gamma_u(c'_u)[\bullet] : \sigma_{c'_u} \to \sigma_j$, where $top_P(o_1 \cdots o_u) : \sigma_j$ (IH). Note that $top_P(o_1 \cdots o_u) = c_u$ in this case, i.e. $\tau_f = \sigma_j$. We let

$$M_u = K[f'V]\,[\pi x/A_u].$$

The shape of M_u then guarantees the right progression in the uth step $o_u p_u$ (after o_u the LTS will reach a configuration of the form $(\gamma_u(c'_u)[f'V], top_P(o_1 \cdots o_u), \cdots)$, from which $p_u = \bar{f'}(A'_u, c'_u)$ can follow).
Because $\nu(V), \nu(\gamma_u(c'_u)) \subseteq \mathsf{Vis}_P(o_1 \cdots o_u)$ and all names introduced in o_u are substituted for above, we have $\nu(M_u) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i) \cup \{f'\}$. By P-visibility, $f' \in \mathsf{Vis}_P(o_1 \cdots o_i)$, so we can conclude that $\nu(M_u) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i)$, i.e. $\nu(\gamma_i(f)) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i)$.

– Suppose now that $d : \sigma_d$ is a continuation name introduced by P in action p_i $(1 \le i \le n)$, or $d = c$, in which case we let $i = 0$. Let us consider all subsequent occurrences of d in t: suppose $\mathcal{I}_d = \{i < u \le n \mid o_u = d(A_u)\}$. Then we let

$$\gamma_i(d) = (\lambda x.\,(time :=!time{+}1); \text{if}\ (!time \in \mathcal{I}_d)\ (assert(x \sim A_{!time}); M_{!time})\ \Omega)[\bullet]$$

where the terms M_u ($u \in \mathcal{I}_c$) are the same as in the previous case, though this time we aim for $x : \tau_d \vdash M_u : \tau_j$, where $d : \tau_d$ and $top_P(o_1 \cdots o_u) : \tau_j$ (recall that $\xi_i(d) = top_P(o_1 \cdots o_u)$). As argued above, in the second case M_u will have the required type and in the first case it can be forced thanks to throw.
Similarly, we can conclude that $\nu(\gamma_i(d)) \subseteq \mathsf{Vis}_P(o_1 \cdots o_i)$.

This completes the definition of configurations. They evolve as required by construction, because the definition of γ_i is compatible with the evolution of the GOSC[HOSC] LTS: at each stage, the value of the clock *time* is incremented and the corresponding term M_u is selected.

It is is easy to check that the syntax used in the construction belongs to GOSC only.

D.4 Proof of Theorem 4

Proof. We follow the same path as in the proof of Theorem 2 except that, in this case, we will have $t, t_1 \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M_1}^{\rho_{\vec{A}_i}, c})$. Consequently, we can conclude that $t_2 = t_1^{\perp}\,\bar{\diamond}((), c')$ is P-visible and invoke Lemma 9 (instead of Lemma 5) to obtain C_O that corresponds to h, K, γ from GOSC. Because k, K, γ are in GOSC, we can then appeal to the assumption $\Gamma \vdash M_1 \lesssim_{err}^{\mathrm{GOSC}(ciu)} M_2$ and complete the proof like for Theorem 2.

E Additional material for Section 5 (HOS[HOSC])

E.1 Proof of Lemma 11

To enable a proof by induction we generalize the Lemma as follows.

Lemma 35. *Consider* $\mathbf{C}_O = \mathsf{C}_{h,K,\gamma}^{\vec{\gamma}_i, c}$*, where* h, K, γ *are from* HOS *and* $(\vec{A}_i, \vec{\gamma}_i) \in \mathbf{AVal}_\Gamma(\gamma)$*. Let* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C}_O)$ *and suppose* $\mathbf{C}_O \xrightarrow{t'} \mathbf{C}$*.*

- *If* t' *is of odd length then* $\mathbf{C} = \langle M, c', \cdots \rangle$ *and* $c' = top_P(t')$*.*
- *If* t' *is of even length and* $t' = t\bar{f}(A, c')$ *then* $\mathbf{C} = \langle \cdots, \xi, \cdots \rangle$ *and* $\xi(c') = top_P(t)$*.*
- *If* t' *is of even length and* $t' = t\bar{c'}(A)$ *then* $c' = top_P(t)$*.*

Proof. By induction on the number of transitions in $\mathbf{C}_O \xrightarrow{t} \mathbf{C}$. In the base case (no transitions) the Lemma holds vacuously.

Note that the Lemma is preserved by silent transitions (t is of odd length then) by Lemma 10.

Suppose $\mathbf{C}_O \xrightarrow{t} \mathbf{C} \xrightarrow{a} \mathbf{C}'$.

- The even-length cases follow immediately the odd-length case due to the shape of LTS rules.
- Suppose $t' = ta$ is of odd length.

- If $a = f(A, c'')$ then $top_P(t') = c''$ and $c' = c''$, so the Lemma holds.
- If $a = c''(A)$ then $c' = \xi(c'')$.
  * If $c'' = c$ then $c' = \circ$ and indeed $top_P(t') = \circ$.
  * Otherwise $top_P(t') = top_P(t'')$, where c'' is introduced by an action (question) after t''. Then, by IH, $\xi(c'') = top_P(t'')$. Because $top_P(t'') = top_P(t')$, we get $c' = top_P(t')$, as required.

E.2 Proof of Theorem 5

Proof. Suppose $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$. Consider Σ, h, K, γ (as in the definition of $\lesssim^{\mathrm{HOS}}_{ciu,err}$) such that $(K[M_1\{\gamma\}], h) \Downarrow_{err}$. In particular, h, K, γ consist of HOS syntax. Suppose $(\vec{A_i}, \vec{\gamma_i}) \in \mathbf{AVal}_\Gamma(\gamma)$ and $c : \sigma$ $(c \neq \circ)$. By Lemma 4 (left-to-right), there exist t, c' such that $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_1})$ and $t^\perp \,\bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i},c}_{h,K,\gamma})$. By Lemma 11, $t^\perp \,\bar{\diamond}((), c')$ is P-bracketed. Thus, t is O-bracketed and, by Lemma 12 (right-to-left), $t \in \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_1})$. From $\mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_1) \subseteq \mathbf{Tr}_{\mathrm{HOS}}(\Gamma \vdash M_2)$, we get $t \in \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$. By Lemma 12 (left-to-right), $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$. Because $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\rho_{\vec{A_i}},c}_{M_2})$ and $t^\perp \,\bar{\diamond}((), c') \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}^{\vec{\gamma_i},c}_{h,K,\gamma})$, by Lemma 4 (right-to-left), we can conclude $(K[M_2\{\gamma\}], h) \Downarrow_{err}$. Thus, $\Gamma \vdash M_1 \lesssim^{\mathrm{HOS}}_{ciu,err} M_2$.

E.3 Proof of Lemma 13

Proof. We take advantage of the definability result for HOSC (Lemma 34) and argue that, for P-bracketed traces, continuation-related syntax can be eliminated. This will follow from the careful integration of $top_P()$ in the construction.

Indeed, the only place where "throw" is needed in the construction is to transition from configuration D_i to E_i. The second component (current continuation) in D_i is equal to $top_P(o_1 \cdots o_{i+1})$, whereas the second component in E_i in this case is $c_O^{j'}$. For a P-bracketed trace, the two continuation names will be the same (Definition 17). Consequently, the use of "throw" in this case is trivial: it will have the form (throw $\mathrm{V_A}$ to cont $(\bullet, c), c, \cdots)$, where $c = c_O^{j'}$, because the continuation in $cor_{j'}$ is cont $(\bullet, c_O^{j'})$ by one of our invariants. This use of "throw" can be replaced simply by $(V_A, c, \dots)$, i.e. occurrences of "throw" can be eliminated.

Next, one observes that references to continuations (cor_j) are redundant as well, because they are only used in connection with "throw", and we already know that "throw" is redundant.

Finally, "callcc" is redundant, because the only purpose of invoking it was to record continuations in a reference, and we know from the previous point that such references will not be needed.

Overall this yields a construction that involves only HOS syntax.

E.4 Proof of Theorem 6

Proof. We follow the same path as in the proof of Theorem 2 except that, in this case, we have $t, t_1 \in \mathbf{Tr}_{\mathrm{HOS}}(\mathsf{C}_{M_1}^{\rho_{\vec{A_i}},c})$. Consequently, we can conclude that $t_2 = t_1^{\perp} \bar{\diamond}((), c')$ is P-bracketed and invoke Lemma **??** (instead of Lemma 5) to obtain C_O that corresponds to h, K, γ from HOS. Because k, K, γ are in HOS, we can appeal to the assumption $\Gamma \vdash M_1 \lesssim_{ciu,err}^{\mathrm{HOS}} M_2$ and complete the proof like for Theorem 2.

F Additional material for Section 6 (GOS[HOSC])

F.1 GOS[HOSC] LTS

$$
\begin{array}{l|lcl}
(P\tau) & \langle M, c, \gamma, \xi, \phi, h, \mathcal{F}\rangle & \xrightarrow{\tau} & \langle N, c', \gamma, \xi, \phi, h', \mathcal{F}\rangle \\
& \text{when } (M, c, h) \to (N, c', h') & & \\
(PA) & \langle V, c, \gamma, \xi, \phi, h, \mathcal{F}\rangle & \xrightarrow{\bar{c}(A)} & \langle \gamma \cdot \gamma', \xi, \phi \uplus \nu(A), h, \mathcal{F}, \mathcal{F}(c) \uplus \nu(A), c'\rangle \\
& \text{when } c : \sigma,\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ \xi(c) = c' & & \\
(PQ) & \langle K[fV], c, \gamma, \xi, \phi, h, \mathcal{F}\rangle & \xrightarrow{\bar{f}(A,c')} & \langle \gamma \cdot \gamma' \cdot [c' \mapsto K], \xi \cdot [c' \mapsto c], \phi \uplus \phi', h, \mathcal{F}, \mathcal{F}(f) \uplus \phi', c'\rangle \\
& \text{when } f : \sigma \to \sigma',\ (A, \gamma') \in \mathbf{AVal}_\sigma(V),\ c' : \sigma' \text{ and } \phi' = \nu(A) \uplus \{c'\} & & \\
(OA) & \langle \gamma, \xi, \phi, h, \mathcal{F}, \mathcal{V}, c''\rangle & \xrightarrow{c(A)} & \langle K[A], c', \gamma, \xi, \phi \uplus \nu(A), h, \mathcal{F} \cdot [\nu(A) \mapsto \mathcal{V}]\rangle \\
& \text{when } c \in \mathcal{V},\ c = c'',\ c : \sigma,\ A : \sigma,\ \gamma(c) = K,\ \xi(c) = c' & & \\
(OQ) & \langle \gamma, \xi, \phi, h, \mathcal{F}, \mathcal{V}, c''\rangle & \xrightarrow{f(A,c)} & \langle VA, c, \gamma, \xi \cdot [c \mapsto c''], \phi \uplus \phi', h, \mathcal{F} \cdot [\phi' \mapsto \mathcal{V}]\rangle \\
& \text{when } f \in \mathcal{V},\ f : \sigma \to \sigma',\ A : \sigma,\ c : \sigma',\ \gamma(f) = V \text{ and } \phi' = \nu(A) \uplus \{c\} & &
\end{array}
$$

Given $N \subseteq$ Names, $[N \mapsto \mathcal{V}]$ stands for the map $[n \mapsto \mathcal{V} \mid n \in N]$.

Fig. 10. GOS[HOSC] LTS

Recall that, given a Γ-assignment ρ, term $\Gamma \vdash M : \tau$ and $c \in \mathrm{CNames}_\tau$, the active configuration $\mathsf{C}_M^{\rho,c}$ was defined by $\mathsf{C}_M^{\rho,c} = \langle M\{\rho\}, c, \emptyset, \emptyset, \nu(\rho) \cup \{c\}, \emptyset\rangle$. We need to upgrade it to the LTS by initializing the new components: $\mathsf{C}_{M,vis,bra}^{\rho,c} = \langle M\{\rho\}, c, \emptyset, [c \mapsto \bot], \nu(\rho) \cup \{c\}, \emptyset, \emptyset\rangle$.

Definition 33. *The* GOS[HOSC] **trace semantics** *of a cr-free HOSC term $\Gamma \vdash M : \tau$ is defined to be* $\mathbf{Tr}_{\mathrm{GOS}}(\Gamma \vdash M : \tau) = \{((\rho, c), t) \mid \rho$ *is a Γ-assignment,* $c : \tau,\ t \in \mathbf{Tr}_{\mathrm{GOSC}}(\mathsf{C}_{M,vis,bra}^{\rho,c})\}$.

By construction and from the GOSC and HOS sections, it follows that

Lemma 36. $t \in \mathbf{Tr}_{\mathrm{GOS}}(\mathsf{C}_{M,vis,bra}^{\rho,c})$ *iff* $t \in \mathbf{Tr}_{\mathrm{HOSC}}(\mathsf{C}_M^{\rho,c})$ *and t is O-visible and O-bracketed.*

Lemma 37 (Definability). *Suppose* $\phi \uplus \{\diamond\} \subseteq \mathrm{FNames}$ *and* t *is an even-length P-bracketed and P-visible* $(\{\circ_{\tau'}, \diamond\}, \phi \uplus \{c\})$*-trace starting with an O-action. There exists a passive configuration* $\mathbf{C}$ *such that the even-length traces* $\mathbf{Tr}_{\mathrm{HOSC}}(\mathbf{C})$ *are exactly the even-length prefixes of* t *(along with all renamings that preserve types and* $\phi \uplus \{c, \circ_{\tau'}, \diamond\}$*). Moreover,* $\mathbf{C} = \langle \gamma \cdot [c \mapsto K], \{c \mapsto \circ_{\tau'}\}, \phi \uplus \{c, \circ_{\tau'}, \diamond\}, h\rangle$*, where* h, K, γ *are built from* GOS *syntax.*

Proof. Follows from the argument for GOSC. We first observe that throw is needed before answer actions to adjust the continuation from $top_O(o_1 \cdots o_i)$. With P-bracketing there is no need for such adjustments. Consequently, we do not need call/cc, which was used to generate continuations to be used in future adjustments.